Package: rmorie (via r-universe)

June 23, 2026

Type Package

Title Multi-Domain Open Research and Inferential Estimation

Version 0.9.5.12

Author Vansh Singh Ruhela [aut, cre] (<https://orcid.org/0009-0004-1750-3592>)

Maintainer Vansh Singh Ruhela <hadesllm@proton.me>

Description Multi-domain scientific computing toolkit for observational inference and intervention analysis across scientific-experimentation contexts, hosting the MRM (Multilevel Reconciliation Methodology) framework for Canadian carceral, police, and oversight data as its primary application. Provides general-purpose causal estimators (ATE, ATT, ATC, GATE, CATE, LATE, AIPW, G-computation), survey sampling methods (stratified, cluster, PPS, bootstrap, calibration weights), propensity-score and doubly-robust estimators, and sensitivity analyses (E-value, Rosenbaum bounds). Companion modules support signal processing and spectral analysis, cryptographic primitives, spatial statistics, statistical physics of crime (Hawkes self-exciting processes, reaction-diffusion, Levy flight, urban scaling), and classical-test-theory and item-response-theory psychometrics, alongside ingestion utilities for officially published Ontario Special Investigations Unit (police-oversight) and federal Structured Intervention Unit reports.

License AGPL (>= 3) + file LICENSE

LazyData true

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.3.0)

Note Tested on R 4.3.0, 4.4.x, 4.5.2, 4.5.3 (CRAN). Use CRAN R on Apple Silicon (Homebrew R may segfault with RSQLite).

Imports here, stats, utils, Rcpp

LinkingTo Rcpp, RcppArmadillo

SystemRequirements libcurl: libcurl-devel (rpm) or libcurl4-openssl-dev (deb)

Suggests AER, AIPW, anominate, arrow, bacondecomp, basicspace, bigrquery, boot, bootstrap, broom, cansim, car, caret, CausalImpact, cmprsk, cobalt, coin, copula, data.table, DBI, dbscan, DescTools, designmatch, did, DIDmultiplegt, digest, dirichletprocess, dlm, DoubleML, DRDID, duckdb, e1071, ebal, eegkit, effects, effectsize, elasticnet, emhawkes, emmeans, EValue, fdrtool, fixest, FNN, gbm, geepack, ggplot2, glmnet, gMCP, gmm, gstat, harmonicmeanp, hawkes, HonestDiD, httr2, iml, ipred, irr, ivreg, jsonlite, kernlab, knitr, konfound, lcmm, lmtest, locfdr, marginaleffects, margins, MASS, Matching, MatchIt, MCMCpack, metafor, mgcv, mlr3, mlr3learners, MSwM, multcomp, mutoss, mvtnorm, nortest, np, openssl, oro.nifti, patchwork, pdftools, performance, pkgload, plm, poolr, ppcor, pracma, pROC, pscl, psych, qvalue, quantreg, R.matlab, randomForest, randtests, randtoolbox, ranger, rbounds, rddensity, rdlocrand, rdpower, rdrobust, readr, readxl, resample, rmarkdown, rmgarch, rmoriedata, rpart, RPostgres, rsample, RSQLite, Rtsne, rugarch, rvest, sandwich, seewave, sensemakr, sensitivitymv, sf, signal, simpleboot, smacof, smotefamily, sodium, spdep, stdReg, survey, survival, synthdid, testthat (>= 3.0.0), tipr, torch, tseries, TwoWayFEWeights, urca, vars, wavelets, WeightIt, withr, writexl, xgboost, xml2

VignetteBuilder knitr

URL https://github.com/rootcoder007/rmorie

BugReports https://github.com/rootcoder007/rmorie/issues

Config/testthat/edition 3

biocViews Software, StatisticalMethod, Survival, Spatial, TimeSeries, Bayesian, MultipleComparison, Visualization

Remotes rootcoder007/rmoriedata, synth-inference/synthdid

Config/pak/sysreqs libssl-dev

Repository https://rootcoder007.r-universe.dev

Date/Publication 2026-06-23 09:44:28 UTC

RemoteUrl https://github.com/rootcoder007/rmorie

RemoteRef main

RemoteSha 0fca70c74a5b201457a4b6ac9d2fe20115579fd0

Contents

adjust_p_values 34

all_stat_command_names . . . 35
analyze_doob_full_affidavit . . . 35
analyze_doob_table1_releases . . . 35
analyze_doob_table2_flow . . . 36
analyze_doob_table3_age_overrepresentation . . . 36
anova_table . . . 37
arsau . . . 37
arsau_analyze . . . 38
arsau_datasets . . . 39
ARSAU_KINDS . . . 40
ARSAU_REGISTRY . . . 40
ARSAU_YEARS . . . 40
assess_calibration . . . 41
assess_discrimination . . . 41
audit_variables . . . 42
auto_test . . . 42
bartlett_test . . . 43
benjamini_hochberg . . . 43
benjamini_yekutieli . . . 44
bias_adjusted_estimate . . . 44
block_bootstrap . . . 45
bonferroni . . . 46
bootstrap . . . 46
bootstrap_632 . . . 47
bootstrap_effect_size_ci . . . 48
bootstrap_validate . . . 49
buttbp . . . 49
buttbs . . . 50
butthp . . . 51
buttlp . . . 52
cauchy_combination . . . 52
CCRSO_TABLE1_RELEASES . . . 53
CCRSO_TABLE2_FLOW . . . 53
CCRSO_TABLE3_AGE . . . 54
cepst . . . 54
cheatsheet . . . 55
check_referential_integrity . . . 55
chi2_goodness_of_fit . . . 56
chi2_independence . . . 56
ckan_metadata . . . 57
cles . . . 57
cliffs_delta . . . 58
cochrans_q . . . 58
coef.crr . . . 59
coefficient_of_variation . . . 59
cohens_f . . . 60
cohens_kappa . . . 60
cohens_w . . . 61

collinearity_diagnostics . . . 61
column_rule . . . 62
commands_by_category . . . 63
compute_goodness_of_fit . . . 63
compute_influence . . . 64
compute_residuals . . . 64
compute_vif . . . 65
correlation_matrix . . . 65
correlation_table . . . 66
create_reproducibility_manifest . . . 66
cross_validate . . . 67
d_to_nnt . . . 68
d_to_or . . . 68
d_to_r . . . 69
dagostino_pearson . . . 69
dataset_catalog . . . 70
decision_curve_analysis . . . 71
decoupling_test . . . 71
delete_d_jackknife . . . 72
detect_overfitting . . . 73
dfa . . . 73
doob_trends . . . 74
e_value . . . 75
e_value_d . . . 75
e_value_hr . . . 76
e_value_or . . . 76
e_value_rr . . . 77
ecgdet . . . 77
effect_size_result . . . 78
effect_sizes . . . 79
effects . . . 79
epsilon_squared . . . 80
estimate_ate . . . 81
estimate_ate_gcomputation . . . 81
estimate_pi0 . . . 82
estimate_pliv . . . 82
estimate_plr . . . 83
explain_file . . . 84
explain_known_files . . . 85
extenders_nonparam . . . 85
extenders_rdd . . . 85
extenders_spatial . . . 86
extenders_stats . . . 86
external_validate . . . 86
fairness_cityprofile . . . 87
fairness_metrics . . . 87
fairness_temporal . . . 88
fallback_procedure . . . 88

fisher_combined . . . 89
fisher_exact_test . . . 89
fixed_effects_meta . . . 90
fixed_sequence . . . 90
format_dataframe . . . 91
format_number . . . 91
friedman_test . . . 92
frns_metrics . . . 92
frns_predpol . . . 93
frns_temporal . . . 94
full_diagnostics . . . 94
gam_smoother . . . 95
glass_delta . . . 96
harmonic_mean_p . . . 96
hazard_ratio_table . . . 97
hcepst . . . 97
hdecon . . . 98
hfd . . . 99
hierarchical_bonferroni . . . 100
hochberg . . . 100
holm . . . 101
holm_sidak . . . 101
hommel . . . 102
hosmer_lemeshow_test . . . 102
hrvfd . . . 103
hrvnl . . . 103
hrvtd . . . 104
i_squared . . . 105
incidence_rate_difference . . . 105
intraclass_correlation . . . 106
jackknife . . . 106
kde . . . 107
kendall_correlation . . . 108
kernel-codes . . . 108
kernel_cond_moments . . . 109
kernel_eval . . . 109
kfd . . . 110
ks_test_one_sample . . . 111
ks_test_two_sample . . . 111
leave_one_out_cv . . . 112
levene_test . . . 112
license_check . . . 113
likelihood_ratio_test . . . 113
lilliefors_test . . . 114
link_test . . . 114
local_fdr . . . 115
local_linear . . . 115
longitudinal_sim . . . 116

loocv_bandwidth . . . 117
mann_whitney_u . . . 117
manski_bounds . . . 118
mcnemar_test . . . 118
midranks . . . 119
model_comparison_table . . . 119
morie_anominate_ideal_points . . . 120
morie_anova_one_way . . . 121
morie_arch_in_mean . . . 121
morie_arsau_analyze_aggregate_summary . . . 122
morie_arsau_analyze_detailed_dataset . . . 123
morie_arsau_analyze_individual_records . . . 124
morie_arsau_analyze_main_records . . . 125
morie_arsau_analyze_probe_cycle_records . . . 126
morie_arsau_analyze_weapon_records . . . 126
morie_arsau_available_datasets . . . 127
morie_arsau_available_years . . . 128
morie_arsau_ckan_url . . . 128
morie_arsau_describe . . . 129
morie_arsau_download . . . 129
morie_arsau_fetch_sidecar . . . 130
morie_arsau_load_aggregate_summary . . . 131
morie_arsau_load_detailed_dataset . . . 131
morie_arsau_load_individual_records . . . 132
morie_arsau_load_main_records . . . 132
morie_arsau_load_probe_cycle_records . . . 133
morie_arsau_load_weapon_records . . . 133
morie_arsau_read_markdown_dictionary . . . 134
morie_arsau_read_sidecar . . . 134
morie_arsau_read_xlsx_dictionary . . . 135
morie_arsau_registry_df . . . 136
morie_arsau_sidecar_schema . . . 136
morie_arsau_sidecar_to_frame . . . 137
morie_audit_all_variables . . . 137
morie_audit_arsau_variables . . . 138
morie_audit_otis_variables . . . 138
morie_audit_public_outputs . . . 139
morie_bayes_cpi_genomic . . . 140
morie_bayes_ridge_gibbs . . . 141
morie_bayesian_lasso_full . . . 142
morie_bayesian_ridge_regression . . . 143
morie_boot_basic_ci . . . 143
morie_boot_run . . . 144
morie_bootstrap_sample . . . 145
morie_build_outputs_manifest . . . 145
morie_builtin_db . . . 146
morie_cache_clear . . . 147
morie_cache_dir . . . 147

morie_cache_file . . . 149
morie_cache_list . . . 149
morie_cache_load . . . 150
morie_cache_store . . . 151
morie_calculate_ebac . . . 151
morie_calculate_ipw_weights . . . 152
morie_calibration_weights . . . 153
morie_canonicalize_cpads_data . . . 154
morie_causal_impact . . . 155
morie_causal_robust_se . . . 156
morie_causal_weighting . . . 157
morie_check_plugin_license . . . 158
morie_chi_square_test . . . 159
morie_ckan_search . . . 159
morie_classify_variable . . . 160
morie_cli_dump_catalog . . . 161
morie_cluster_sample . . . 161
morie_cnn_genomic . . . 162
morie_coherence . . . 163
morie_coin_independence . . . 163
morie_coin_oneway . . . 164
morie_coin_wilcoxon . . . 165
morie_compare_nested_logistic_models . . . 165
morie_compute_design_weights . . . 166
morie_concordance_incomplete . . . 167
morie_confusion_matrix_metrics . . . 167
morie_contingency_coefficient . . . 168
morie_control_comparison . . . 169
morie_control_median_test . . . 169
morie_copula_fit . . . 170
morie_copula_sample . . . 171
morie_corrections_uof_resource_ids . . . 171
morie_cpads_canonicalize_frame . . . 172
morie_cpads_contract . . . 173
morie_cpads_has_raw_columns . . . 173
morie_cpads_infer_file_format . . . 174
morie_cpads_missing_variables . . . 174
morie_cpads_validate_frame . . . 175
morie_crypto_chacha20_poly1305_decrypt . . . 176
morie_crypto_chacha20_poly1305_encrypt . . . 176
morie_crypto_hkdf_sha256 . . . 177
morie_crypto_hybrid_decrypt . . . 178
morie_crypto_hybrid_encrypt . . . 178
morie_crypto_hybrid_keygen . . . 179
morie_crypto_keystore_create . . . 179
morie_crypto_keystore_list . . . 180
morie_crypto_keystore_load . . . 180
morie_crypto_keystore_store . . . 181

morie_crypto_liboqs_available . . . 181
morie_crypto_liboqs_version . . . 182
morie_crypto_mldsa65_keygen . . . 182
morie_crypto_mldsa65_sign . . . 183
morie_crypto_mldsa65_verify . . . 183
morie_crypto_mlkem768_decaps . . . 184
morie_crypto_mlkem768_encaps . . . 184
morie_crypto_mlkem768_keygen . . . 185
morie_crypto_random_bytes . . . 185
morie_crypto_sodium_available . . . 186
morie_crypto_sodium_version . . . 186
morie_dataset_catalog . . . 187
morie_dataset_column_profile . . . 187
morie_dataset_detect_role . . . 188
morie_dataset_infer_level . . . 189
morie_dataset_info . . . 190
morie_dataset_load . . . 190
morie_dataset_portal_catalog . . . 191
morie_dataset_portal_catalog_clear_cache . . . 192
morie_dataset_profile . . . 192
morie_dataset_profile_summary_table . . . 193
morie_dataset_profile_to_list . . . 194
morie_dataset_suggest_plan . . . 194
morie_dataset_summarize_column . . . 195
morie_datasets_arcgis_item_by_id . . . 195
morie_datasets_arcgis_item_metadata . . . 196
morie_datasets_arsau_aggregate_summary . . . 197
morie_datasets_arsau_detailed_dataset . . . 197
morie_datasets_arsau_uof_individual_records . . . 198
morie_datasets_arsau_uof_main_records . . . 198
morie_datasets_arsau_uof_probe_cycle_records . . . 199
morie_datasets_arsau_uof_weapon_records . . . 200
morie_datasets_bigquery . . . 201
morie_datasets_browse . . . 201
morie_datasets_calgary_community_crime_stats . . . 203
morie_datasets_calgary_open_crime_adjacent_layers . . . 203
morie_datasets_calgary_socrata_by_id . . . 204
morie_datasets_chicago_arrests . . . 204
morie_datasets_chicago_community_areas . . . 206
morie_datasets_chicago_crime . . . 207
morie_datasets_chicago_crime_map . . . 209
morie_datasets_chicago_crime_odata . . . 210
morie_datasets_chicago_crime_resolved . . . 212
morie_datasets_chicago_crime_soql . . . 214
morie_datasets_chicago_iucr_codes . . . 215
morie_datasets_chicago_neighborhoods . . . 216
morie_datasets_chicago_police_beats . . . 218
morie_datasets_chicago_police_districts . . . 219

morie_datasets_chicago_wards . . . 221
morie_datasets_ckan_package . . . 222
morie_datasets_ckan_search . . . 223
morie_datasets_corrections_uof_ethnic_origin . . . 223
morie_datasets_corrections_uof_incident_type . . . 224
morie_datasets_corrections_uof_incidents . . . 224
morie_datasets_corrections_uof_indigenous . . . 225
morie_datasets_corrections_uof_inmate_incident . . . 225
morie_datasets_corrections_uof_inmate_participant . . . 226
morie_datasets_corrections_uof_institution_summary . . . 226
morie_datasets_corrections_uof_location_summary . . . 227
morie_datasets_corrections_uof_race . . . 227
morie_datasets_corrections_uof_religion . . . 228
morie_datasets_corrections_uof_select_incident_summary . . . 228
morie_datasets_corrections_uof_staff_incident . . . 229
morie_datasets_cpads . . . 229
morie_datasets_cpd_public_arrests . . . 231
morie_datasets_edmonton_police_stations . . . 232
morie_datasets_external_socrata_layers . . . 233
morie_datasets_load_by_key . . . 234
morie_datasets_montreal_ckan_resource . . . 235
morie_datasets_montreal_justice_safety_layers . . . 236
morie_datasets_montreal_sim_intervention_types . . . 237
morie_datasets_montreal_sim_interventions . . . 237
morie_datasets_namus_missing_persons . . . 238
morie_datasets_nibrs . . . 239
morie_datasets_nist_rds . . . 240
morie_datasets_nyc_boroughs . . . 240
morie_datasets_nyc_boundaries_catalog . . . 241
morie_datasets_nyc_community_districts . . . 242
morie_datasets_nyc_council_districts . . . 243
morie_datasets_nyc_ntas_2020 . . . 243
morie_datasets_nyc_nypd_arrests_historic . . . 244
morie_datasets_nyc_nypd_arrests_ytd . . . 245
morie_datasets_nyc_nypd_boro_crosswalk . . . 246
morie_datasets_nyc_nypd_by_key . . . 246
morie_datasets_nyc_nypd_complaint_historic . . . 247
morie_datasets_nyc_nypd_complaint_ytd . . . 248
morie_datasets_nyc_nypd_hate_crimes . . . 249
morie_datasets_nyc_nypd_law_books . . . 250
morie_datasets_nyc_nypd_layers . . . 251
morie_datasets_nyc_nypd_offense_codes . . . 251
morie_datasets_nyc_nypd_resolved . . . 252
morie_datasets_nyc_nypd_uof_incidents . . . 254
morie_datasets_nyc_nypd_uof_subjects . . . 255
morie_datasets_nyc_nypd_vehicle_stops . . . 256
morie_datasets_nyc_opendata_bulk_layers . . . 257
morie_datasets_nyc_police_precincts . . . 258

morie_datasets_nyc_school_districts . . . 259
morie_datasets_nyc_socrata_by_id . . . 259
morie_datasets_nyc_stop_and_frisk . . . 260
morie_datasets_nyc_zctas . . . 261
morie_datasets_ontario_ckan_by_key . . . 261
morie_datasets_ontario_ckan_layers . . . 262
morie_datasets_otis_a01 . . . 262
morie_datasets_otis_a01_restrictive_confinement . . . 263
morie_datasets_otis_b01_segregation_detailed . . . 264
morie_datasets_otis_b02_segregation_total_days . . . 264
morie_datasets_otis_b03_seg_alerts_by_institution . . . 265
morie_datasets_otis_b04_seg_consecutive_by_region . . . 266
morie_datasets_otis_b05_seg_consecutive_lengths . . . 266
morie_datasets_otis_b06_seg_reason_by_institution . . . 267
morie_datasets_otis_b07_seg_alerts_by_gender . . . 268
morie_datasets_otis_b08_seg_consecutive_by_institution . . . 268
morie_datasets_otis_b09_seg_n_times . . . 269
morie_datasets_otis_c01_individuals_total . . . 270
morie_datasets_otis_c02_individuals_by_institution . . . 270
morie_datasets_otis_c03_individuals_race_by_gender . . . 271
morie_datasets_otis_c04_individuals_race_by_region . . . 272
morie_datasets_otis_c05_individuals_religion_by_region . . . 272
morie_datasets_otis_c06_individuals_age_by_region . . . 273
morie_datasets_otis_c07_individuals_alerts . . . 274
morie_datasets_otis_c08_individuals_religion_by_gender . . . 274
morie_datasets_otis_c09_individuals_age_by_gender . . . 275
morie_datasets_otis_c10_aggregate_durations_by_institution . . . 276
morie_datasets_otis_c11_aggregate_lengths . . . 276
morie_datasets_otis_c12_aggregate_durations_by_region . . . 277
morie_datasets_otis_d01_deaths_in_custody . . . 278
morie_datasets_otis_d02_deaths_by_gender . . . 279
morie_datasets_otis_d03_deaths_by_race . . . 279
morie_datasets_otis_d04_deaths_by_religion . . . 280
morie_datasets_otis_d05_deaths_by_age_category . . . 281
morie_datasets_otis_d06_cause_by_alert . . . 281
morie_datasets_otis_d07_alerts_by_housing_unit . . . 282
morie_datasets_siu_director_reports . . . 283
morie_datasets_siu_report_fields . . . 283
morie_datasets_siu_report_text . . . 284
morie_datasets_statcan_ccjs_cubes . . . 284
morie_datasets_statcan_cube_metadata . . . 285
morie_datasets_statcan_full_csv_url . . . 285
morie_datasets_statcan_vectors . . . 286
morie_datasets_summary . . . 287
morie_datasets_toronto_ambulance_stations . . . 287
morie_datasets_toronto_asr_miscellaneous . . . 288
morie_datasets_toronto_open_ckan_resource . . . 288
morie_datasets_toronto_open_crime_adjacent_layers . . . 289

morie_datasets_toronto_zoning_per_neighbourhood . . . 289
morie_datasets_tps_2008_firs . . . 290
morie_datasets_tps_2009_firs . . . 291
morie_datasets_tps_2010_firs . . . 292
morie_datasets_tps_2011_firs . . . 293
morie_datasets_tps_2012_firs . . . 294
morie_datasets_tps_2013_firs . . . 295
morie_datasets_tps_administrative . . . 296
morie_datasets_tps_arcgis_hub_by_id . . . 297
morie_datasets_tps_arcgis_hub_download . . . 299
morie_datasets_tps_arcgis_hub_layers . . . 299
morie_datasets_tps_arrested_and_charged_persons . . . 301
morie_datasets_tps_arrests_and_strip_searches . . . 302
morie_datasets_tps_assault . . . 303
morie_datasets_tps_automobile_ksi . . . 303
morie_datasets_tps_autotheft . . . 304
morie_datasets_tps_bicycle_thefts . . . 305
morie_datasets_tps_bicycletheft . . . 306
morie_datasets_tps_breakandenter . . . 307
morie_datasets_tps_budget_2020 . . . 307
morie_datasets_tps_budget_2021 . . . 308
morie_datasets_tps_budget_2022 . . . 309
morie_datasets_tps_budget_2023 . . . 310
morie_datasets_tps_budget_2024 . . . 311
morie_datasets_tps_budget_2025 . . . 312
morie_datasets_tps_budget_2026 . . . 313
morie_datasets_tps_budget_by_command . . . 314
morie_datasets_tps_calls_for_service_attended . . . 315
morie_datasets_tps_community_safety_indicators . . . 316
morie_datasets_tps_complaint_dispositions . . . 317
morie_datasets_tps_cyclist_ksi . . . 318
morie_datasets_tps_dispatched_calls_by_division . . . 319
morie_datasets_tps_facilities . . . 320
morie_datasets_tps_fatals_ksi . . . 321
morie_datasets_tps_firearms_top_calibres . . . 322
morie_datasets_tps_gross_expenditures_by_division . . . 323
morie_datasets_tps_gross_operating_budget . . . 324
morie_datasets_tps_hatecrimes . . . 325
morie_datasets_tps_homicide . . . 326
morie_datasets_tps_homicides . . . 326
morie_datasets_tps_intimate_partner_family_violence . . . 327
morie_datasets_tps_investigated_alleged_misconduct . . . 328
morie_datasets_tps_killed_and_seriously_injured . . . 329
morie_datasets_tps_layers . . . 330
morie_datasets_tps_major_crime . . . 330
morie_datasets_tps_mha_apprehensions . . . 331
morie_datasets_tps_miscellaneous_calls_for_service . . . 332
morie_datasets_tps_miscellaneous_data . . . 333

morie_datasets_tps_miscellaneous_firearms . . . 334
morie_datasets_tps_motorcylist_ksi . . . 335
morie_datasets_tps_neighbourhood_crime_rates . . . 336
morie_datasets_tps_passenger_ksi . . . 337
morie_datasets_tps_patrol_zone . . . 338
morie_datasets_tps_pedestrian_ksi . . . 339
morie_datasets_tps_personnel_by_command . . . 340
morie_datasets_tps_personnel_by_rank . . . 341
morie_datasets_tps_personnel_by_rank_by_division . . . 342
morie_datasets_tps_persons_in_crisis_calls_for_service_attended . . . 343
morie_datasets_tps_police_divisions . . . 344
morie_datasets_tps_psdp_resolved . . . 345
morie_datasets_tps_regulated_interactions . . . 346
morie_datasets_tps_reported_crimes . . . 347
morie_datasets_tps_robbery . . . 348
morie_datasets_tps_search_of_persons . . . 348
morie_datasets_tps_shooting_firearm_discharges . . . 349
morie_datasets_tps_shootings . . . 350
morie_datasets_tps_staffing_by_command . . . 351
morie_datasets_tps_theft_from_motor_vehicle . . . 352
morie_datasets_tps_theft_over . . . 352
morie_datasets_tps_tickets_issued . . . 353
morie_datasets_tps_top_20_offences_of_firearm_seizures . . . 354
morie_datasets_tps_total_public_complaints . . . 355
morie_datasets_tps_traffic_collisions . . . 356
morie_datasets_tps_use_of_force_call_for_service_types . . . 357
morie_datasets_tps_use_of_force_call_sources_by_month . . . 358
morie_datasets_tps_use_of_force_gender_composition . . . 359
morie_datasets_tps_use_of_force_location_of_occurrences . . . 360
morie_datasets_tps_use_of_force_occurrence_category . . . 361
morie_datasets_tps_use_of_force_time_of_day_trends . . . 362
morie_datasets_tps_use_of_force_use_of_force_types_and_perceived_weapons 363
morie_datasets_tps_victims_of_crime . . . 364
morie_datasets_vancouver_opendata_by_id . . . 365
morie_datasets_vancouver_opendata_layers . . . 366
morie_datasets_vpd_crime . . . 366
morie_datasets_vpd_legal_disclaimer . . . 369
morie_db_connect . . . 369
morie_db_create_indexes . . . 370
morie_dbscan_clustering . . . 371
morie_dcc_multivariate_garch . . . 371
morie_decision_tree_split . . . 372
morie_deep_learning_genomic . . . 373
morie_default_synthetic_name_map . . . 374
morie_default_workflow_map . . . 374
morie_desc_atkinson . . . 375
morie_desc_cramers_v . . . 375
morie_desc_gini . . . 376

morie_desc_kappa . . . 376
morie_desc_winsorize . . . 377
morie_describe . . . 377
morie_describe_by_name . . . 378
morie_design_effect . . . 379
morie_det_rng . . . 379
morie_det_rng_sha_hex . . . 380
morie_did_2x2 . . . 381
morie_did_aggregate_gt_att . . . 382
morie_did_bacon_decomposition . . . 383
morie_did_chaisemartin_dhaultfoeuille . . . 383
morie_did_continuous_treatment . . . 384
morie_did_diagnostics . . . 385
morie_did_doubly_robust . . . 386
morie_did_event_study . . . 387
morie_did_fuzzy . . . 388
morie_did_group_time_att . . . 389
morie_did_heterogeneous . . . 390
morie_did_panel_fe . . . 391
morie_did_parallel_trends_data . . . 392
morie_did_placebo_test_group . . . 392
morie_did_placebo_test_outcome . . . 393
morie_did_placebo_test_time . . . 394
morie_did_repeated_cross_section . . . 395
morie_did_sensitivity_analysis . . . 396
morie_did_staggered . . . 397
morie_did_synthdid_estimate . . . 398
morie_did_synthetic . . . 399
morie_did_test_parallel_trends . . . 400
morie_did_triple_difference . . . 401
morie_did_twoway_fe_weights . . . 402
morie_did_wild_cluster_bootstrap . . . 403
morie_diffu_diffusion_forward . . . 404
morie_download_bootstrap . . . 405
morie_dp_gaussian_mixture . . . 406
morie_dsp_acf_from_psd . . . 407
morie_dsp_alpha_trimmed_mean . . . 407
morie_dsp_amplitude_histogram . . . 408
morie_dsp_arc_length . . . 408
morie_dsp_band_power . . . 409
morie_dsp_baseline_correlation . . . 409
morie_dsp_centroidal_time . . . 410
morie_dsp_coherence . . . 410
morie_dsp_coherence_spectrum . . . 411
morie_dsp_comb . . . 412
morie_dsp_complex_cepstrum . . . 412
morie_dsp_complex_demodulation . . . 413
morie_dsp_crest_factor . . . 414

morie_dsp_cross_correlation 414
morie_dsp_csd 415
morie_dsp_cv 415
morie_dsp_derivative_detect 416
morie_dsp_dicrotic_notch 417
morie_dsp_ensemble_average 417
morie_dsp_entropy_histogram 418
morie_dsp_even_odd 418
morie_dsp_fbm_synthesis 419
morie_dsp_form_factor 419
morie_dsp_fractal_dim_psd 420
morie_dsp_hann_filter 420
morie_dsp_higuchi_fd 421
morie_dsp_hilbert_envelope 422
morie_dsp_hjorth 422
morie_dsp_hjorth_activity 423
morie_dsp_hjorth_complexity 423
morie_dsp_hjorth_mobility 424
morie_dsp_homomorphic 424
morie_dsp_hr_from_rr 425
morie_dsp_integrated_emg 425
morie_dsp_katz_fd 426
morie_dsp_lms 426
morie_dsp_matched 427
morie_dsp_mean_abs 427
morie_dsp_mean_frequency 428
morie_dsp_median_filter 428
morie_dsp_median_frequency 429
morie_dsp_min_phase 430
morie_dsp_moving_average 430
morie_dsp_myopulse_rate 431
morie_dsp_nlms 431
morie_dsp_notch 432
morie_dsp_onset_detect 433
morie_dsp_pan_tompkins 433
morie_dsp_parzen_pdf 434
morie_dsp_psd_bartlett 435
morie_dsp_psd_periodogram 435
morie_dsp_psd_to_db 436
morie_dsp_psd_welch 436
morie_dsp_qrs_features 437
morie_dsp_rls 438
morie_dsp_rms 438
morie_dsp_ruler_fd 439
morie_dsp_shannon_energy 439
morie_dsp_shape_factor 440
morie_dsp_slope_sign_changes 440
morie_dsp_snr 441

morie_dsp_snr_improvement 442
morie_dsp_spectral_edge 442
morie_dsp_spectral_entropy 443
morie_dsp_spectral_flatness 443
morie_dsp_spectral_kurtosis 444
morie_dsp_spectral_moment 444
morie_dsp_spectral_ratio 445
morie_dsp_synchronized_average 446
morie_dsp_t_wave 446
morie_dsp_teager_energy 447
morie_dsp_template_match 447
morie_dsp_threshold_detect 448
morie_dsp_turning_points 449
morie_dsp_turns_count 449
morie_dsp_variance_ratio 450
morie_dsp_waveform_length 450
morie_dsp_waveform_length_norm 451
morie_dsp_wiener_filter 451
morie_dsp_wiener_hopf 452
morie_dsp_willison_amplitude 452
morie_dsp_window 453
morie_dsp_zero_crossing 454
morie_e_value 454
morie_effects_comparisons 455
morie_effects_emmeans 456
morie_effects_predictions 456
morie_effects_slopes 457
morie_effects_tidy 457
morie_eg_coint 458
morie_egarch_model 458
morie_ensure_extras 459
morie_entheo_analyze_subject 460
morie_entheo_available_subjects 461
morie_entheo_clone_dmt_imaging 461
morie_entheo_dataset_overview 462
morie_entheo_dynamic_functional_connectivity 463
morie_entheo_load_eeg_region 463
morie_entheo_load_fmri_subject 464
morie_entheo_lz_complexity 464
morie_entheo_spectral_band_power 465
morie_estimate_aipw 466
morie_estimate_atc 467
morie_estimate_ate 468
morie_estimate_att 469
morie_estimate_cate 469
morie_estimate_double_ml 471
morie_estimate_g_computation 472
morie_estimate_gate 473

morie_estimate_irm . . . 474
morie_estimate_late . . . 476
morie_estimate_propensity_scores . . . 477
morie_ewma_volatility . . . 478
morie_fairness_apply_profile . . . 478
morie_fairness_average_odds_difference . . . 479
morie_fairness_bias_amplification . . . 480
MORIE_FAIRNESS_CANONICAL_FIELDS . . . 481
morie_fairness_city_profile . . . 482
morie_fairness_column_map . . . 483
morie_fairness_demographic_parity . . . 483
morie_fairness_disparate_impact . . . 484
morie_fairness_equalized_odds . . . 485
morie_fairness_get_city . . . 487
morie_fairness_gini . . . 487
morie_fairness_list_cities . . . 488
morie_fairness_noisy_or_detection . . . 488
morie_fairness_predpol . . . 489
morie_fairness_predpol_aggregate_areas . . . 489
morie_fairness_predpol_calibration_audit . . . 490
morie_fairness_predpol_score_disparity . . . 491
morie_fairness_predpol_temporal_audit . . . 491
morie_fairness_register_city . . . 492
morie_fairness_simulate_biased_crime_data . . . 493
morie_fairness_simulation . . . 494
morie_fairness_xai . . . 494
morie_fairness_xai_ale . . . 494
morie_fairness_xai_ceteris_paribus . . . 495
morie_fairness_xai_partial_dependence . . . 496
morie_fairness_xai_permutation_importance . . . 496
morie_fairness_xai_shap_values . . . 497
morie_fast_available . . . 498
morie_fdr_qvalues . . . 499
morie_fetch . . . 499
morie_fetch_arcgis . . . 501
morie_fetch_ckan . . . 502
morie_fetch_siu . . . 503
morie_fetch_tps . . . 504
morie_fisher_exact_test . . . 505
morie_garch_fit . . . 506
morie_gblup_full . . . 507
morie_generate_ar_coefficients . . . 507
morie_generate_synthetic_data . . . 508
morie_generate_var_coefficients . . . 509
morie_genomic_cross_validation . . . 510
morie_geostat_krige . . . 510
morie_geostat_variogram . . . 511
morie_ghosal_adaptation . . . 512

morie_ghosal_bernstein_von_mises . . . 513
morie_ghosal_contraction_rate . . . 514
morie_ghosal_dirichlet_posterior . . . 514
morie_ghosal_dpmixture_density . . . 515
morie_ghosal_empirical_bayes . . . 516
morie_ghosal_gp_matern . . . 516
morie_ghosal_gp_squared_exponential . . . 517
morie_ghosal_hierarchical_bayes . . . 518
morie_ghosal_log_density . . . 519
morie_ghosal_moment_matching . . . 519
morie_ghosal_neutral_right . . . 520
morie_ghosal_np_classification . . . 521
morie_ghosal_np_regression . . . 522
morie_ghosal_np_testing . . . 522
morie_ghosal_posterior_consistency . . . 523
morie_ghosal_sieve_prior . . . 524
morie_ghosal_stick_breaking_trunc . . . 524
morie_ghosal_survival_beta_process . . . 525
morie_ghosal_wavelet_prior . . . 526
morie_gpl_compatible_licenses . . . 526
morie_gradient_boosting_ensemble . . . 527
morie_gradient_boosting_genomic . . . 528
morie_gradient_descent_vanilla . . . 529
morie_grid_search_cv . . . 529
morie_grm_vanraden . . . 530
morie_gxe_interaction_model . . . 531
morie_hawkes_fit . . . 532
morie_hurst_r . . . 533
morie_infer_measurement_level . . . 533
morie_ingest_bigquery_build_sql . . . 534
morie_ingest_bigquery_query . . . 535
morie_ingest_bigquery_table . . . 536
morie_ingest_chicago_crime . . . 538
morie_ingest_chicago_crime_bigquery . . . 539
morie_ingest_chicago_resources . . . 540
morie_ingest_chicago_socrata . . . 540
morie_ingest_cihi_xlsx . . . 541
morie_ingest_ckan_fetch_package_csvs . . . 542
morie_ingest_ckan_package_search . . . 543
morie_ingest_ckan_package_show . . . 544
morie_ingest_ckan_read_resource . . . 544
morie_ingest_ckan_resource_show . . . 545
morie_ingest_ckan_search_packages . . . 546
morie_ingest_forensics_namus_missing . . . 547
morie_ingest_forensics_nibrs . . . 548
morie_ingest_forensics_nist_rds . . . 549
morie_ingest_statcan_cansim . . . 550
morie_ingest_statcan_csv . . . 551

morie_ingest_tps_feature_layer . . . 552
morie_ingest_tps_fetch . . . 553
morie_ingest_tps_layers . . . 554
morie_inspect_output . . . 554
morie_install_extras . . . 555
morie_is_over_legal_limit . . . 557
morie_iv_anderson_rubin . . . 557
morie_iv_anderson_rubin_ci . . . 558
morie_iv_conditional_lr . . . 559
morie_iv_control_function . . . 560
morie_iv_cragg_donald . . . 560
morie_iv_cue_gmm . . . 561
morie_iv_diagnostics . . . 562
morie_iv_durbin_wu_hausman . . . 563
morie_iv_first_stage_diagnostics . . . 563
morie_iv_gmm . . . 564
morie_iv_hansen_j . . . 565
morie_iv_hausman . . . 565
morie_iv_jive . . . 566
morie_iv_kleibergen_paap . . . 566
morie_iv_liml . . . 567
morie_iv_panel . . . 568
morie_iv_probit . . . 568
morie_iv_residual_analysis . . . 569
morie_iv_sargan . . . 570
morie_iv_split_sample . . . 570
morie_iv_stock_yogo . . . 571
morie_iv_tsls . . . 571
morie_iv_wald . . . 572
morie_jackknife_estimate . . . 573
morie_johansen_cointegration . . . 574
morie_kalman_filter . . . 574
morie_kendall_tau . . . 575
morie_kendall_tau_partial . . . 576
morie_kernel_pca . . . 576
morie_kmeans_clustering . . . 577
morie_kruskal_wallis_test . . . 578
morie_laniyonu_actuarial_risk_disparity . . . 578
morie_laniyonu_gentrification_policing . . . 580
morie_laniyonu_smi_force_disparity . . . 582
morie_lcmm_latent_class . . . 584
morie_learning_curve . . . 585
morie_levene_test . . . 585
morie_license_metadata . . . 586
morie_linear_regression_ols . . . 586
morie_list_datasets . . . 587
morie_list_morie_modules . . . 588
morie_llm_agent_available . . . 588

morie_llm_ask . . . 589
morie_llm_ask_multi . . . 589
morie_llm_detect_provider . . . 590
morie_llm_list_freeapi_models . . . 591
morie_llm_probe_freeapi . . . 591
morie_llm_probe_ollama . . . 592
morie_llm_request_completion . . . 592
morie_load_cpads . . . 593
morie_load_cpads_data . . . 594
morie_load_dataset . . . 594
morie_locfdr_estimate . . . 595
morie_mann_whitney_test . . . 596
morie_marker_variance . . . 597
morie_matching_abadie_imbens_se . . . 598
morie_matching_atc_matched . . . 599
morie_matching_ate_matched . . . 599
morie_matching_att_matched . . . 600
morie_matching_balance . . . 601
morie_matching_balance_table . . . 602
morie_matching_cardinality . . . 603
morie_matching_cem . . . 604
morie_matching_common_support . . . 605
morie_matching_doubly_robust . . . 606
morie_matching_entropy_balance . . . 607
morie_matching_estimate_propensity . . . 608
morie_matching_exact . . . 609
morie_matching_full . . . 609
morie_matching_genetic . . . 610
morie_matching_longitudinal . . . 611
morie_matching_love_plot_data . . . 612
morie_matching_mahalanobis . . . 613
morie_matching_multi_treatment . . . 614
morie_matching_nearest_neighbor . . . 615
morie_matching_optimal_pair . . . 616
morie_matching_overlap . . . 617
morie_matching_quality . . . 617
morie_matching_rosenbaum_bounds . . . 618
morie_matching_subclassify . . . 619
morie_matching_trim_propensity . . . 620
morie_matching_variable_ratio . . . 621
morie_meta_rma . . . 622
morie_midas_regression . . . 623
morie_mini_batch_gradient . . . 623
morie_ml_apply_smote . . . 624
morie_ml_eval_robustness . . . 625
morie_multi_trait_gblup . . . 625
morie_multiple_testing . . . 626
morie_mvn_with_covariance . . . 627

morie_mvnorm_pmv . . . 628
morie_mvnorm_sample . . . 629
morie_nbeats_basis . . . 629
morie_np_kernel_reg . . . 630
morie_nyc_nypd_socrata_cap_note . . . 631
morie_odds_ratio_ci . . . 632
morie_omega_squared . . . 632
morie_one_sample_coverage . . . 633
morie_one_sample_t_test . . . 633
morie_ordered_alternatives_test . . . 634
morie_ordered_categories . . . 635
morie_otis_aipw_ate . . . 635
morie_otis_aipw_superlearner . . . 637
morie_otis_all_analyses . . . 637
morie_otis_all_analyze . . . 638
morie_otis_analyze_a01 . . . 639
morie_otis_analyze_a01_ruhela_alt_age . . . 639
morie_otis_analyze_a01_ruhela_alt_gender . . . 640
morie_otis_analyze_a01_ruhela_alt_toronto . . . 641
morie_otis_analyze_a01_ruhela_formulations . . . 641
morie_otis_analyze_a01_ruhela_per_year . . . 642
morie_otis_analyze_a01_ruhela_subgroup_female . . . 643
morie_otis_analyze_a01_ruhela_subgroup_male . . . 643
morie_otis_analyze_a01_with_csi_context . . . 644
morie_otis_analyze_all . . . 645
morie_otis_analyze_b01 . . . 645
morie_otis_analyze_b01_ruhela_alt_age . . . 646
morie_otis_analyze_b01_ruhela_alt_gender . . . 646
morie_otis_analyze_b01_ruhela_alt_toronto . . . 647
morie_otis_analyze_b01_ruhela_formulations . . . 647
morie_otis_analyze_b01_ruhela_per_year . . . 648
morie_otis_analyze_b01_ruhela_subgroup_female . . . 649
morie_otis_analyze_b01_ruhela_subgroup_male . . . 649
morie_otis_analyze_b02 . . . 650
morie_otis_analyze_b02_ruhela_alt_age . . . 650
morie_otis_analyze_b02_ruhela_alt_region . . . 651
morie_otis_analyze_b02_ruhela_formulations . . . 651
morie_otis_analyze_b03 . . . 652
morie_otis_analyze_b03_ruhela_aggregate . . . 652
morie_otis_analyze_b04 . . . 653
morie_otis_analyze_b04_ruhela_aggregate . . . 653
morie_otis_analyze_b05 . . . 654
morie_otis_analyze_b05_mandela_classification . . . 654
morie_otis_analyze_b05_ruhela_aggregate . . . 655
morie_otis_analyze_b06 . . . 655
morie_otis_analyze_b06_ruhela_aggregate . . . 656
morie_otis_analyze_b07 . . . 656
morie_otis_analyze_b07_ruhela_aggregate . . . 657

morie_otis_analyze_b08 . . . 657
morie_otis_analyze_b08_ruhela_aggregate . . . 658
morie_otis_analyze_b09 . . . 658
morie_otis_analyze_b09_ruhela_aggregate . . . 659
morie_otis_analyze_c_chi2 . . . 659
morie_otis_analyze_c01 . . . 660
morie_otis_analyze_c01_ruhela_aggregate . . . 660
morie_otis_analyze_c01_ruhela_aggregate_region_cluster . . . 661
morie_otis_analyze_c02 . . . 662
morie_otis_analyze_c02_ruhela_aggregate . . . 662
morie_otis_analyze_c03 . . . 663
morie_otis_analyze_c03_ruhela_aggregate . . . 663
morie_otis_analyze_c04 . . . 664
morie_otis_analyze_c04_ruhela_aggregate . . . 664
morie_otis_analyze_c04_ruhela_aggregate_region_cluster . . . 665
morie_otis_analyze_c05 . . . 665
morie_otis_analyze_c05_ruhela_aggregate . . . 666
morie_otis_analyze_c06 . . . 666
morie_otis_analyze_c06_ruhela_aggregate . . . 667
morie_otis_analyze_c07 . . . 667
morie_otis_analyze_c07_ruhela_aggregate . . . 668
morie_otis_analyze_c08 . . . 668
morie_otis_analyze_c08_ruhela_aggregate . . . 669
morie_otis_analyze_c09 . . . 669
morie_otis_analyze_c09_ruhela_aggregate . . . 670
morie_otis_analyze_c10 . . . 670
morie_otis_analyze_c10_ruhela_aggregate . . . 671
morie_otis_analyze_c11 . . . 671
morie_otis_analyze_c11_mandela_classification . . . 672
morie_otis_analyze_c11_ruhela_aggregate . . . 672
morie_otis_analyze_c12 . . . 673
morie_otis_analyze_c12_ruhela_aggregate . . . 673
morie_otis_analyze_d_chi2 . . . 674
morie_otis_analyze_d01 . . . 674
morie_otis_analyze_d02 . . . 675
morie_otis_analyze_d02_ruhela_aggregate . . . 675
morie_otis_analyze_d03 . . . 676
morie_otis_analyze_d03_ruhela_aggregate . . . 676
morie_otis_analyze_d04 . . . 677
morie_otis_analyze_d04_ruhela_aggregate . . . 677
morie_otis_analyze_d05 . . . 678
morie_otis_analyze_d05_ruhela_aggregate . . . 678
morie_otis_analyze_d06 . . . 679
morie_otis_analyze_d07 . . . 679
morie_otis_analyze_otis_mandela_provincial_vs_federal . . . 680
morie_otis_analyze_ruhela_grid . . . 680
morie_otis_analyze_ruhela_master . . . 681
morie_otis_analyze_ruhela_per_year . . . 682

morie_otis_analyzers . . . 683
morie_otis_astcmb . . . 683
morie_otis_causal_grid . . . 684
morie_otis_churn . . . 684
morie_otis_churn_analyze_all . . . 685
morie_otis_classify_mandela_combo . . . 686
morie_otis_disciplinary_medical_overlap . . . 687
morie_otis_embedding_distribution . . . 688
morie_otis_intra_year_transition_matrix . . . 688
morie_otis_ipw_ate . . . 689
morie_otis_irm_dml . . . 690
morie_otis_irr_glmm_vm . . . 691
morie_otis_load . . . 692
morie_otis_make_pair_a . . . 693
morie_otis_make_pair_alert_to_volatility_a01 . . . 693
morie_otis_make_pair_alert_to_volatility_all . . . 694
morie_otis_make_pair_alert_to_volatility_naive . . . 695
morie_otis_make_pair_alert_to_volatility_ruhela . . . 695
morie_otis_make_pair_b . . . 696
morie_otis_make_pair_c . . . 697
morie_otis_mortification_cooccurrence . . . 697
morie_otis_otdesc . . . 698
morie_otis_otdml . . . 698
morie_otis_path_complexity_gini . . . 699
morie_otis_pipeline . . . 700
morie_otis_plr . . . 700
morie_otis_primitives . . . 701
morie_otis_psm . . . 702
morie_otis_psm_subclass . . . 703
morie_otis_rctrnd . . . 703
morie_otis_regC_demog_contingency . . . 704
morie_otis_region_alert_state_richness . . . 704
morie_otis_repeat_placement_concentration . . . 705
morie_otis_rplace . . . 705
morie_otis_tps_analyze_all . . . 706
morie_otis_tps_composite_overlay . . . 707
morie_otis_tps_overlay . . . 707
morie_otis_tps_per_region_rollup . . . 708
morie_otis_tps_yoy_correlation . . . 708
morie_otis_volat . . . 709
morie_otis_within_year_placement_count . . . 710
morie_otis_within_year_region_diversity . . . 710
morie_paired_t_test . . . 711
morie_parse_nypd_law_code . . . 711
morie_paths . . . 712
morie_pca_dimension_reduction . . . 713
morie_pcg_filter . . . 713
morie_penalized_regression . . . 714

morie_percentile_modified_rank 715
morie_performance_check_collinearity 716
morie_performance_check_model 716
morie_performance_check_outliers 717
morie_performance_r2 717
morie_point_biserial_r 718
morie_polynomial_regression 718
morie_power_prop_test 719
morie_power_t_test 720
morie_ppcor_partial 721
morie_ppcor_semipartial 721
morie_pps_sample 722
morie_prediction_accuracy 723
morie_predpol_aggregate_areas 723
morie_predpol_calibration_audit 724
morie_predpol_score_disparity 725
morie_predpol_temporal_audit 726
morie_profile_dataset 727
morie_prophet_components 727
morie_proportion_ci 728
morie_psymet_alpha 729
morie_psymet_alphadel 729
morie_psymet_ave 730
morie_psymet_bartlett 730
morie_psymet_cr 731
morie_psymet_discrimination 731
morie_psymet_itemtotal 732
morie_psymet_kmo 732
morie_psymet_omega 733
morie_psymet_parallel 733
morie_psymet_splithalf 734
morie_quantile_reg 734
morie_random_forest_ensemble 735
morie_random_forest_genomic 736
morie_random_search_cv 737
morie_randtests_bartels 738
morie_randtests_runs 738
morie_randtests_turning_point 739
morie_rank_based_test 739
morie_rank_order_statistics 740
morie_rank_placements 741
morie_rdd_bandwidth_cct 741
morie_rdd_bandwidth_ik 742
morie_rdd_bandwidth_rot 742
morie_rdd_bandwidth_sensitivity 743
morie_rdd_bias_corrected 743
morie_rdd_cattaneo_density 744
morie_rdd_covariate_balance 745

morie_rdd_density_test . . . 746
morie_rdd_discrete . . . 746
morie_rdd_donut . . . 747
morie_rdd_fuzzy . . . 748
morie_rdd_geographic . . . 749
morie_rdd_kernels . . . 750
morie_rdd_kink . . . 750
morie_rdd_local_polynomial . . . 751
morie_rdd_local_randinf . . . 752
morie_rdd_local_randomisation . . . 753
morie_rdd_mccrary . . . 753
morie_rdd_placebo_cutoff . . . 754
morie_rdd_plot_data . . . 755
morie_rdd_power . . . 756
morie_rdd_power_calc . . . 756
morie_rdd_sample_size . . . 757
morie_rdd_sharp . . . 758
morie_read_outputs_manifest . . . 759
morie_recommended_pair_test . . . 760
morie_recommended_summary . . . 760
morie_regime_switching . . . 761
morie_regularization_path . . . 761
morie_risk_difference_ci . . . 762
morie_risk_ratio_ci . . . 763
morie_rkhs_full . . . 763
morie_rnn_genomic . . . 764
morie_roc_auc_score . . . 765
morie_rsample_bootstraps . . . 765
morie_run_ebac_selection_ipw_analysis . . . 766
morie_run_morie_module . . . 767
morie_run_morie_modules . . . 768
morie_run_pipeline . . . 769
morie_run_propensity_ipw_analysis . . . 770
morie_run_treatment_effects_analysis . . . 771
morie_run_weighted_logistic_analysis . . . 772
morie_run_workflow_step . . . 773
morie_sample . . . 774
morie_sample_size_logistic . . . 774
morie_semipar_bridge . . . 775
morie_sensitivity_evalue . . . 776
morie_sensitivity_konfound . . . 777
morie_sensitivity_omitted_var_bias . . . 778
morie_sensitivity_rosenbaum . . . 779
morie_sensitivity_tipping_point . . . 780
morie_sgolay_smooth . . . 780
morie_shapiro_wilk_test . . . 781
morie_sign_test_power . . . 782
morie_simple_random_sample . . . 782

morie_simpleboot_two . . . 783
morie_simulate_longitudinal_panel . . . 784
morie_siu_all_analyses . . . 785
morie_siu_analyze . . . 786
morie_siu_anomaly_check . . . 786
morie_siu_audit_case . . . 787
morie_siu_audit_columns . . . 788
morie_siu_by_police_service . . . 790
morie_siu_by_year . . . 791
morie_siu_cache_path . . . 791
morie_siu_case_counts . . . 792
morie_siu_charges_by_year_chi2 . . . 792
morie_siu_classify_mandela . . . 793
morie_siu_compare . . . 794
morie_siu_decision_timing . . . 795
morie_siu_demographics . . . 796
morie_siu_fetch . . . 796
morie_siu_fetch_cases . . . 797
morie_siu_fetch_dataframe . . . 798
morie_siu_index . . . 799
morie_siu_llm_extract . . . 800
morie_siu_mental_health_race_indicators . . . 801
morie_siu_parse_html . . . 802
morie_siu_parse_news_html . . . 802
morie_siu_parser . . . 803
morie_siu_record_correction . . . 803
morie_siu_refresh_manifest . . . 804
morie_siu_sanity_check . . . 806
morie_siu_sprott_doob_feb2021 . . . 807
morie_siu_sprott_doob_iftene_table1 . . . 807
morie_siu_sprott_doob_iftene_table10 . . . 808
morie_siu_sprott_doob_iftene_table15 . . . 808
morie_siu_sprott_doob_iftene_table9 . . . 809
morie_siu_sprott_doob_table11 . . . 809
morie_siu_sprott_doob_table12 . . . 809
morie_siu_sprott_doob_table13 . . . 810
morie_siu_sprott_doob_table15 . . . 810
morie_siu_sprott_doob_table19 . . . 811
morie_siu_sprott_doob_table22 . . . 811
morie_siu_sprott_doob_table23 . . . 812
morie_siu_sprott_doob_table4 . . . 812
morie_siu_translate . . . 813
morie_siu_verify_chi2 . . . 815
morie_siu_verify_published_chi_squares . . . 815
MORIE_SIUIAP_AFFIDAVITS . . . 816
morie_siuiap_cite . . . 816
MORIE_SIUIAP_CRIMSL_REPORTS . . . 817
MORIE_SIUIAP_ORIGINAL_PANEL_2019_2020 . . . 817

MORIE_SIUIAP_PANEL_MANDATE 818
MORIE_SIUIAP_PANEL_MEMBERS 818
morie_siuiap_panel_summary 818
MORIE_SIUIAP_REPORTS 819
MORIE_SIUIAP_URL 819
morie_spatial_voting_aldrich_mckelvey 820
morie_spatial_voting_alpha_nominate 821
morie_spatial_voting_anchoring_vignettes 822
morie_spatial_voting_bayesian_am 822
morie_spatial_voting_bayesian_irt_likelihood 823
morie_spatial_voting_bayesian_irt_posterior 824
morie_spatial_voting_bayesian_mds 825
morie_spatial_voting_bayesian_unfolding 826
morie_spatial_voting_blackbox 826
morie_spatial_voting_cjr_irt 827
morie_spatial_voting_classical_mds 828
morie_spatial_voting_cutting_lines 829
morie_spatial_voting_double_centering 829
morie_spatial_voting_dw_nominate 830
morie_spatial_voting_dynamic_irt 831
morie_spatial_voting_em_irt 832
morie_spatial_voting_ideal_point_recovery 832
morie_spatial_voting_indscal 833
morie_spatial_voting_mds_fit_stats 834
morie_spatial_voting_mlsmu6 835
morie_spatial_voting_nominate_bootstrap 836
morie_spatial_voting_nominate_loglik 837
morie_spatial_voting_nominate_utility 838
morie_spatial_voting_nominate_vote_prob 838
morie_spatial_voting_nonmetric_mds 839
morie_spatial_voting_nonparametric_bootstrap 840
morie_spatial_voting_normal_vectors 841
morie_spatial_voting_optimal_classification 842
morie_spatial_voting_ordered_oc 843
morie_spatial_voting_ordinal_irt 843
morie_spatial_voting_procrustes 844
morie_spatial_voting_smacof 845
morie_spatial_voting_smacof_unfolding 846
morie_spatial_voting_unfolding_stress 847
morie_spatial_voting_wordfish 847
morie_spearman_rho 848
morie_specs_from_df 849
morie_spectral_cluster 849
morie_spectral_density 850
morie_sprott_doob 851
morie_stat_bridge 851
morie_stat_commands 852
morie_state_space_model 852

morie_stratified_sample 853
morie_suggest_analysis_plan 854
morie_sukhatme_test 854
morie_summarize_output_audit 855
morie_survey_calibrate 855
morie_survey_complex_glm 856
morie_survey_design 857
morie_survey_glm 857
morie_survey_hajek_mean 858
morie_survey_ht_total 858
morie_survey_mean 859
morie_survey_poststratify 859
morie_survey_ratio 860
morie_survey_subpop 860
morie_survival_aft 861
morie_survival_cif 861
morie_survival_compare_parametric 862
morie_survival_concordance 862
morie_survival_cox 863
morie_survival_coxsnell 863
morie_survival_deviance 864
morie_survival_finegray 864
morie_survival_hr 865
morie_survival_km 865
morie_survival_landmark 866
morie_survival_left_truncated_km 866
morie_survival_logrank 867
morie_survival_martingale 867
morie_survival_nelsonaalen 868
morie_survival_parametric 868
morie_survival_rmst 869
morie_survival_rmst_diff 869
morie_survival_schoenfeld 870
morie_survival_turnbull 870
morie_svm_genomic 871
morie_svm_hinge_primal 871
morie_svm_kernel_trick 872
morie_sync_rng 873
morie_synth_corrections_uof 873
morie_synth_otis 874
morie_synth_otis_all 875
morie_terry_hoeffding_test 876
morie_tgarch_model 876
morie_threshold_autoregression 877
morie_to_hood_crosswalk 877
morie_to_neighbourhoods 878
morie_tolerance_limits 879
morie_toronto_neighbourhoods 880

morie_tps_add_hood_140_from_158 . . . 880
morie_tps_add_hood_158_from_140 . . . 881
morie_tps_aggregate_158_to_140 . . . 882
morie_tps_analyze . . . 883
morie_tps_analyze_all . . . 884
morie_tps_analyze_assault . . . 884
morie_tps_analyze_csi_from_dataframes . . . 885
morie_tps_analyze_one . . . 886
morie_tps_arima_forecast . . . 886
morie_tps_assert_hood_version . . . 887
morie_tps_available_formats . . . 887
morie_tps_bivariate_moran . . . 888
morie_tps_bivariate_morans_i . . . 889
morie_tps_category_correlation_matrix . . . 890
morie_tps_changepoint_detection . . . 890
morie_tps_compare_hawkes_kernels . . . 891
morie_tps_composite_index . . . 892
morie_tps_crime_compare . . . 893
morie_tps_criminal_network_graph . . . 893
MORIE_TPS_CSI_CATEGORIES . . . 894
morie_tps_csi_per_neighbourhood . . . 895
morie_tps_csi_per_year . . . 895
morie_tps_csi_weight . . . 897
morie_tps_data_dir . . . 897
morie_tps_dbscan_clusters . . . 898
morie_tps_disaggregate_140_to_158 . . . 899
morie_tps_district_for_centroid . . . 900
morie_tps_fetch_category . . . 900
morie_tps_fetch_dataframe . . . 901
morie_tps_fokker_planck_grid . . . 902
morie_tps_getis_ord_g_star . . . 902
morie_tps_gini_concentration . . . 903
morie_tps_hawkes_advanced . . . 904
morie_tps_hawkes_advanced_fit . . . 905
morie_tps_hawkes_markovian_vs_nonmarkovian . . . 906
morie_tps_hawkes_temporal_fit . . . 907
morie_tps_inspection_game_phase . . . 908
morie_tps_kde_density . . . 909
morie_tps_langevin_simulate . . . 910
morie_tps_layer_urls . . . 910
morie_tps_levy_flight_alpha . . . 911
morie_tps_list_categories . . . 912
morie_tps_list_datasets . . . 912
morie_tps_list_formats . . . 913
morie_tps_load . . . 913
morie_tps_load_dataset . . . 914
morie_tps_local_morans_i . . . 915
morie_tps_lotka_volterra_police_crime . . . 916

morie_tps_moran_sweep_heatmap . . . 917
morie_tps_morans_i_neighbourhood . . . 918
morie_tps_neighbourhood_concentration . . . 919
morie_tps_offence_summary . . . 920
morie_tps_polygon_morans_i . . . 920
morie_tps_pretty_label . . . 921
morie_tps_project_xy . . . 922
morie_tps_psdp_layers . . . 922
MORIE_TPS_REGISTRY . . . 923
morie_tps_render . . . 923
morie_tps_render_choropleth . . . 924
morie_tps_render_dbscan . . . 925
morie_tps_render_district_proportional . . . 926
morie_tps_render_points . . . 926
morie_tps_render_quad . . . 927
morie_tps_render_satscan_panel . . . 928
morie_tps_render_yearly_grid . . . 928
morie_tps_resolve_hood_col . . . 929
morie_tps_ripley_k . . . 930
morie_tps_sarima_forecast . . . 931
morie_tps_sdb_reaction_diffusion . . . 931
morie_tps_sdb_turing_demo . . . 933
morie_tps_seasonal_pattern . . . 934
morie_tps_spatial_summary . . . 934
morie_tps_statphysics . . . 935
morie_tps_statphysics_analyze_all . . . 936
morie_tps_stochastic . . . 937
MORIE_TPS_SUPPORTED_FORMATS . . . 937
morie_tps_temporal . . . 938
morie_tps_temporal_summary . . . 938
MORIE_TPS_TORONTO_POPULATION_BY_YEAR . . . 939
MORIE_TPS_TOTAL_CSI_WEIGHTS . . . 939
morie_tps_urban_scaling_beta . . . 940
morie_tps_use_of_force . . . 941
MORIE_TPS_VIOLENT_CSI_WEIGHTS . . . 942
morie_tps_year_over_year_trend . . . 942
morie_tps_year_to_hood_version . . . 943
morie_tps_yoy_panel . . . 943
morie_transformer_genomic . . . 944
morie_tsne_reduction . . . 945
morie_two_sample_coverage . . . 946
morie_two_sample_t_test . . . 946
morie_unobserved_components . . . 947
morie_userguide . . . 948
morie_validate_cpads_data . . . 948
morie_validate_outputs_manifest . . . 949
morie_van_der_waerden_test . . . 949
morie_vecm . . . 950

morie_verify_statistical_output . . . 950
morie_vertex_access_token . . . 951
morie_vertex_ask_gemini . . . 952
morie_vertex_health_check . . . 953
morie_vertex_resolve_config . . . 953
morie_vpd_download_instructions . . . 953
morie_wavelet_time_series . . . 954
morie_weights_bootstrap . . . 955
morie_weights_brr . . . 955
morie_weights_calibrate_to_totals . . . 956
morie_weights_combined . . . 956
morie_weights_deff . . . 957
morie_weights_design . . . 957
morie_weights_detect_extreme . . . 958
morie_weights_diagnostics . . . 958
morie_weights_ess . . . 959
morie_weights_fay_brr . . . 959
morie_weights_greg . . . 960
morie_weights_jackknife . . . 960
morie_weights_multiframe . . . 961
morie_weights_nonresponse . . . 961
morie_weights_normalize . . . 962
morie_weights_poststratify . . . 962
morie_weights_propensity_nonresponse . . . 963
morie_weights_rake . . . 963
morie_weights_replicate_variance . . . 964
morie_weights_sdr . . . 965
morie_weights_smooth . . . 965
morie_weights_trim . . . 966
morie_wilcoxon_power . . . 966
morie_wilcoxon_signed_rank_test . . . 967
morie_write_audit_markdown . . . 968
morie_write_synthetic_data . . . 968
morie_xgboost_objective . . . 969
mrm_anova_bonferroni . . . 970
mrm_anova_oneway . . . 971
mrm_anova_power . . . 972
mrm_arsau . . . 973
mrm_assumptions_check . . . 973
mrm_causal_design . . . 974
mrm_check_balancing . . . 975
mrm_check_overlap . . . 976
mrm_classify_mandela . . . 977
mrm_clt_demo . . . 979
mrm_design . . . 980
mrm_diagnostics . . . 980
mrm_doe . . . 981
mrm_factorial_2k . . . 982

mrm_fractional_factorial 983
mrm_gentrification 984
mrm_gentrification_panel 984
mrm_graeco_latin 986
mrm_kulldorff 987
mrm_latin_square 987
mrm_lisa 988
mrm_mathstats 989
mrm_mc_power 989
mrm_median_causal_effect 990
mrm_morans_i 991
mrm_oneprop_test 992
mrm_otis 993
mrm_otis_mandela_spectrum 994
mrm_otis_mortification_cooccurrence 995
mrm_otis_placement_concentration 996
mrm_otis_region_locality 997
mrm_otis_seg_duration_km 998
mrm_perm_block 999
mrm_pit 1000
mrm_primitives_ordinal 1001
mrm_qq_plot 1002
mrm_random_latin 1003
mrm_rcbd 1003
mrm_response_surface 1004
mrm_samples 1005
mrm_siu 1006
mrm_siu_case_to_decision_km 1006
mrm_siu_outcome_classifier 1008
mrm_siu_per_service_rate 1008
mrm_spatial_spillover 1009
mrm_spatial_spillover_decomposition 1010
mrm_standardised_difference 1011
mrm_synthetic_area_exposure 1012
mrm_synthetic_exposure 1013
mrm_threshold_coefficient 1014
mrm_threshold_specific_ordinal 1014
mrm_tps 1016
mrm_tps_kulldorff_scan 1017
mrm_tps_levy_scaling 1018
mrm_tps_lisa 1019
mrm_tps_load_hawkes_refit 1020
mrm_tps_moran_clustering 1020
mrm_tps_neighbourhood_recurrence_km 1021
mrm_tps_polygon_moran_per_year 1022
mrm_two_treatment_test 1023
mrm_twoprop_test 1024
mrm_uof 1025

mrm_uof_data_quality_audit . . . 1025
mrm_uof_demographic_disparity . . . 1026
mrm_uof_force_concentration . . . 1026
mrm_uof_region_locality . . . 1027
mrm_uof_weapon_diversity . . . 1028
mrm_uof_yoy_change . . . 1028
mrm_var_test . . . 1029
n_effective_tests . . . 1030
n_stat_commands . . . 1030
nested_cross_validate . . . 1031
normality_suite . . . 1032
number_needed_to_harm . . . 1033
number_needed_to_treat . . . 1033
nw_regression . . . 1034
odds_ratio . . . 1034
odds_ratio_table . . . 1035
omega_squared . . . 1035
omitted_variable_bias . . . 1036
one_proportion_ztest . . . 1037
one_sample_ttest . . . 1037
one_way_anova . . . 1038
or_to_d . . . 1038
or_to_r . . . 1039
paired_permutation_test . . . 1039
paired_ttest . . . 1040
parametric_bootstrap . . . 1040
partial_correlation . . . 1041
partial_eta_squared . . . 1042
pburg . . . 1042
pcgenv . . . 1043
pcgmur . . . 1044
pcgseg . . . 1044
pearson_correlation . . . 1045
permutation_fdr . . . 1046
permutation_fwer . . . 1046
permutation_test . . . 1047
pettitt_changepoint . . . 1048
pfd . . . 1049
ph_assumption_test . . . 1050
phi_coefficient . . . 1050
point_biserial_correlation . . . 1051
prediction_interval . . . 1051
print.morie_audit_result . . . 1052
print.morie_tps_result . . . 1052
print.morie_variable_taxonomy . . . 1053
probabilistic_bias_analysis . . . 1053
r_effect_size . . . 1054
r_squared . . . 1054

r_to_d . . . 1055
r_to_or . . . 1055
ramsey_reset_test . . . 1056
random_effects_meta . . . 1056
rank_biserial_correlation . . . 1057
rate_ratio . . . 1057
register_stat_command . . . 1058
regression_table . . . 1058
repeated_cv . . . 1059
repeated_measures_anova . . . 1060
resolve_stat_command . . . 1060
risk_difference . . . 1061
risk_ratio . . . 1061
rosenbaum_bounds . . . 1062
rrint . . . 1062
run_stat_command . . . 1063
runs_test . . . 1064
sampen . . . 1064
score_data_quality . . . 1065
score_test . . . 1065
semi_partial_correlation . . . 1066
SemiparKernels . . . 1066
sensitivity . . . 1067
sensitivity_rosenbaum . . . 1067
sensitivity_summary . . . 1068
sgolay . . . 1069
sidak . . . 1070
silverman_bandwidth . . . 1070
simes_combined . . . 1071
spearman_correlation . . . 1071
specification_curve . . . 1071
standardized_coefficients . . . 1072
stat_bridge_exec . . . 1073
stat_bridge_fn_info . . . 1073
stat_bridge_fn_search . . . 1074
stat_bridge_help . . . 1074
stat_bridge_main . . . 1075
stat_bridge_registry_json . . . 1075
stat_bridge_verify . . . 1076
stat_command . . . 1076
statistics . . . 1077
storey_q . . . 1078
stouffer_combined . . . 1078
subsampling . . . 1079
substance_categories . . . 1079
summary_statistics_table . . . 1080
table1 . . . 1081
tables_pub . . . 1082

temporal_validate 1082
tippett_combined 1083
tipping_point_analysis 1083
tps_crime 1084
tps_csi 1085
tps_spatial 1086
tps_spatial_advanced 1086
treatment_effect_table 1087
two_proportion_ztest 1087
two_sample_ttest 1088
two_way_anova 1088
validate_schema 1089
validation 1089
vancouver_crime_adjacent 1090
vargha_delaney_a 1091
variable_taxonomy 1092
variance_equality_suite 1092
variance_ratio 1093
wald_test 1093
welch 1094
welch_ttest 1095
wilcoxon_signed_rank 1095
wild_bootstrap 1096

Index **1097**

`adjust_p_values` *Convenience dispatcher for p-value adjustment methods*

Description

Front-end over `stats::p.adjust` and the rmorie wrappers above. Preserves the rmorie API used by `stat_commands`.

Usage

```
adjust_p_values(p_values, method = "bh", alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`method`	One of `"bonferroni"`, `"sidak"`, `"holm"`, `"hochberg"`, `"hommel"`, `"holm_sidak"`, `"bh"` / `"benjamini_hochberg"` / `"fdr"`, `"by"` / `"benjamini_yekutieli"`, `"storey"`, or `"fwer"` (alias of holm).
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

`all_stat_command_names`
Sorted vector of all command names + aliases

Description

Sorted vector of all command names + aliases

Usage

```
all_stat_command_names()
```

`analyze_doob_full_affidavit`
Replicate Doob's full Federal Court affidavit (Tables 1-3)

Description

Renders Tables 1, 2 and 3 in a single roll-up `morie_result`. Figures 1-4 (time-series CANSIM data) are out-of-scope here; pass your own series to `decoupling_test()` once they are available.

Usage

```
analyze_doob_full_affidavit()
```

Value

A `morie_result` named-list.

`analyze_doob_table1_releases`
Analyse Doob Affidavit Table 1 — 5-year average annual releases

---

Description

Renders Table 1 and computes overall success / revocation rates.

Usage

```
analyze_doob_table1_releases()
```

Value

A `morie_result` named-list with title, summary_lines, tables, interpretation, and payload.

`analyze_doob_table2_flow`
Analyse Doob Affidavit Table 2 — prisoner flow

Description

Renders Table 2 plus year-over-year changes and 5-year averages.

Usage

```
analyze_doob_table2_flow()
```

Value

A `morie_result` named-list.

`analyze_doob_table3_age_overrepresentation`
Analyse Doob Affidavit Table 3 — age over-/under-representation

---

Description

Renders Table 3 plus age-group IRRs for CSC custody and admissions vs Canadian adult population.

Usage

```
analyze_doob_table3_age_overrepresentation()
```

Value

A `morie_result` named-list.

`anova_table` *ANOVA table from a fitted model*

Description

Uses `stats::anova` for sequential (Type-I) tests, or `car::Anova` for Type-II/III if **car** is installed.

Usage

```
anova_table(
  model,
  typ = 2L,
  digits = 3L,
  output_format = "dataframe",
  title = "ANOVA Table"
)
```

Arguments

`model`	An `lm`/`aov`/`glm` fit.
`typ`	ANOVA type (1, 2, 3).
`digits`	Decimal places.
`output_format`	Output target.
`title`	Title.

`arsau` *ARSAU dataset loaders + registry (R-side mirror of morie.arsau_datasets)*

Description

ARSAU = the Ontario Ministry of the Solicitor General's provincial release of Police Use-of-Force incident records (formally "Race-Based and Identity-Based Data on Police Use of Force in Ontario"). Published on the Ontario Data Catalogue at https://data.ontario.ca/dataset/police-use-of-force-race-based-data.

Details

This file ships the R-side equivalents of the Python `morie.arsau_datasets` module:

- `ARSAU_REGISTRY()`: returns the registered (year x kind) entries as a list of lists.

- `morie_arsau_load_main_records()`, `morie_arsau_load_individual_records()`, `morie_arsau_load_pro`
  `morie_arsau_load_weapon_records()`, `morie_arsau_load_aggregate_summary()`,
  `morie_arsau_load_detailed_dataset()`: per-record-type loaders, returning a named list with `data`, `schema`, `sidecar`, `year`, `kind`, `language`, `is_valid`, `n_rows`, `n_cols`, `interpretation`.
- `morie_arsau_available_years()`, `morie_arsau_available_datasets()`, `morie_arsau_describe()`: discovery callables.

Path portability:

No path on the maintainer's workstation is hard-coded. All file resolution goes through `.morie_resolve_arsau_dir` (defined below), which honours, in order:

1. an explicit `data_dir =` argument
2. the `MORIE_ARSAU_DIR` environment variable
3. `MORIE_DATA_DIR/arsau`
4. `morie_cache_dir("arsau")` (only if already populated by a previous `morie_arsau_download()` call – never auto-created at read-time, per CRAN policy)
5. `system.file("extdata", "arsau", package = "rmorie")` – the bundled tiny fixture for unit tests + tutorials
6. stop with a remediation paragraph

2023 weapon-records invalidity gate:

The 2023 release ships `uof_weapon_records_invaliddata.csv`, flagged by the ministry as non-compliant. `morie_arsau_load_weapon_records(2023)` signals an error unless the caller passes `allow_invalid = TRUE`; when allowed, the returned object's `is_valid` field is `FALSE` and its `warnings` list opens with an explicit caveat paragraph.

`arsau_analyze` *Per-record-type ARSAU analysis pipelines (R-side mirror of* `morie.arsau_analyze`*).*

Description

Each public callable in this file loads one ARSAU dataset via the `morie_arsau_load_*` loaders defined in `R/arsau.R` and chains the jurisdiction-agnostic MRM Use-of-Force primitives from `R/mrm_uof.R` over it, returning a single named-list result (classed `c("morie_arsau_result", "morie_rich_result", "list")`) that bundles the loaded data, every sub-analysis, and a multi-paragraph natural-language interpretation.

Details

These analyzers do NOT invent new statistical methods. They wire the generic `mrm_uof_*` callables against the column names that the Ontario open-data release actually publishes. If the upstream schema changes, the generic callables in `R/mrm_uof.R` continue to work; only the column-name constants below need patching.

Public callables:

- morie_arsau_analyze_main_records (per year, 2023 / 2024)
- morie_arsau_analyze_individual_records (per year, 2023 / 2024)
- morie_arsau_analyze_probe_cycle_records (per year, 2023 / 2024)
- morie_arsau_analyze_weapon_records (per year, 2023 / 2024; 2023 requires allow_invalid = TRUE)
- morie_arsau_analyze_aggregate_summary (2020-2022)
- morie_arsau_analyze_detailed_dataset (2020-2022)

Every analyzer returns a list whose named slots (data, sidecar, force_concentration, data_quality, disparity_by_race, ...) hold the constituent sub-results, so callers can drill into individual tests without re-running the full pipeline.

References

Ontario Ministry of the Solicitor General. *Annual Report on Special and Adaptive Units / Data on Police Use of Force in Ontario*: 2020-2022, 2023, and 2024 releases. https://data.ontario.ca/dataset/police-use-of-force-race-based-data. Technical notes accompanying each annual release describe the data-quality reasons for the 2023 weapon-records invalidity flag.

---

arsau_datasets *ARSAU CKAN sidecar helpers + tidy registry view (R-side companion to* morie.arsau_datasets*).*

Description

The main R-side loaders + registry list-of-lists already live in R/arsau.R (morie_arsau_load_main_records(), morie_arsau_load_individual_records(), morie_arsau_load_probe_cycle_records(), morie_arsau_load_weapon_records(), morie_arsau_load_aggregate_summary(), morie_arsau_load_detaile plus ARSAU_REGISTRY(), ARSAU_YEARS(), ARSAU_KINDS(), morie_arsau_read_sidecar(), morie_arsau_available_*(), and morie_arsau_describe()). This file does NOT duplicate those. It adds the remaining surface from the Python source:

Details

- morie_arsau_registry_df: the registry as a tidy data.frame (one row per published file), the canonical R equivalent of ARSAU_REGISTRY in Python.
- morie_arsau_sidecar_schema: simplified [name, type, notes] extraction from a CKAN sidecar (mirrors morie.arsau_datasets.sidecar_schema).
- morie_arsau_sidecar_to_frame: convert the array-of-arrays records body of a CKAN sidecar to a data.frame keyed by fields[].id (mirrors morie.arsau_datasets.sidecar_to_frame)
- morie_arsau_read_xlsx_dictionary: read an Ontario-Catalogue XLSX data-dictionary sidecar (requires **readxl**; gated with requireNamespace).
- morie_arsau_read_markdown_dictionary: read an Ontario-Catalogue Markdown data-dictionary sidecar with no extra dependencies.
- morie_arsau_ckan_url + morie_arsau_fetch_sidecar: build the upstream CKAN datastore_search URL and (optionally) fetch the sidecar JSON via **httr2**.

References

Ontario Ministry of the Solicitor General. *Data on Police Use of Force in Ontario*, 2020-2022 / 2023 / 2024 releases. Published on the Ontario Data Catalogue: https://data.ontario.ca/dataset/police-use-of-force-race-based-data. CKAN `datastore_search` endpoint: `datastore_search` (https://data.ontario.ca/). Each annual release ships per-resource technical notes; the 2023 weapon_records release is explicitly flagged as containing non-compliant data and the open-data file is renamed accordingly.

`ARSAU_KINDS` *Known ARSAU dataset kinds.*

Description

Known ARSAU dataset kinds.

Usage

```
ARSAU_KINDS()
```

`ARSAU_REGISTRY` *Return the ARSAU registry as a list of entries.*

Description

Each entry is itself a named list with `year_or_range`, `kind`, `csv_filename`, `sidecar_filename`, expected row / column counts, `is_valid`, and bilingual descriptions.

Usage

```
ARSAU_REGISTRY()
```

Value

Named list-of-lists.

`ARSAU_YEARS` *Known ARSAU year/range keys.*

Description

Known ARSAU year/range keys.

Usage

```
ARSAU_YEARS()
```

assess_calibration *Comprehensive calibration assessment for binary outcomes*

Description

Comprehensive calibration assessment for binary outcomes

Usage

```
assess_calibration(y_true, y_pred, n_groups = 10L)
```

Arguments

y_true	Integer 0/1 vector.
y_pred	Predicted probabilities.
n_groups	Hosmer-Lemeshow groups.

assess_discrimination *Discrimination assessment for binary classifier*

Description

Discrimination assessment for binary classifier

Usage

```
assess_discrimination(
  y_true,
  y_pred,
  y_pred_ref = NULL,
  n_bootstrap = 1000L,
  confidence = 0.95,
  random_state = 42L
)
```

Arguments

y_true	Integer 0/1 vector.
y_pred	Predicted probabilities.
y_pred_ref	Optional reference-model probabilities for NRI/IDI.
n_bootstrap	Bootstrap reps for AUC CI.
confidence	Confidence level.
random_state	Seed.

audit_variables *Comprehensive per-variable audit walker (R-side mirror)*

Description

R counterpart of `morie.audit_variables`. Walks every column in every OTIS and ARSAU dataset known to the package, classifies each variable via `morie_classify_variable`, and returns a single audit object (or pair of objects, when domain = "both") summarising coverage, levels of measurement, roles, cross-year-safety, and recommended methods per variable.

Details

Pure R; no C/C++ hot path needed (taxonomy is regex + lookup, not CPU-bound). Per `[[feedback_r_cpp_first]]` we'd reach for Rcpp only if profiling showed a real bottleneck.

Public callables:

- `morie_audit_otis_variables`
- `morie_audit_arsau_variables`
- `morie_audit_all_variables`
- `morie_write_audit_markdown`

auto_test *Automatic test selection*

Description

Decision logic:

1. If `y` is NULL, one-sample t-test against zero.
2. If `paired = TRUE`, paired t-test if differences are normal, otherwise Wilcoxon signed-rank.
3. If two independent samples, check both-normal (Shapiro-Wilk for n<=5000, otherwise D'Agostino-Pearson); if both normal, run Student's or Welch's t depending on Levene's test; otherwise Mann-Whitney U.

Usage

```
auto_test(x, y = NULL, paired = FALSE, confidence = 0.95)
```

Arguments

`x`	Numeric vector.
`y`	Optional second sample.
`paired`	Whether samples are paired.
`confidence`	Confidence level.

bartlett_test — *Bartlett's test for equality of variances*

Description

Bartlett's test for equality of variances

Usage

```
bartlett_test(...)
```

Arguments

...	Two or more numeric vectors.

benjamini_hochberg — *Benjamini-Hochberg FDR control*

Description

Thin wrapper over `stats::p.adjust(method = "BH")`.

Usage

```
benjamini_hochberg(p_values, alpha = 0.05, labels = NULL)

bh(p_values, alpha = 0.05, labels = NULL)
```

Arguments

p_values	Numeric vector of raw p-values.
alpha	Significance level.
labels	Optional character vector of test labels.

benjamini_yekutieli *Benjamini-Yekutieli FDR control under arbitrary dependence*

Description

Thin wrapper over `stats::p.adjust(method = "BY")`.

Usage

```
benjamini_yekutieli(p_values, alpha = 0.05, labels = NULL)

by_fdr(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

bias_adjusted_estimate *Bias-adjusted treatment effect (Ding & VanderWeele 2016)*

Description

Bias-adjusted treatment effect (Ding & VanderWeele 2016)

Usage

```
bias_adjusted_estimate(estimate, se, rr_ud, rr_eu, prevalence_confounder = 0.5)
```

Arguments

`estimate`	Observed treatment effect on the log-RR / coefficient scale.
`se`	Standard error.
`rr_ud`	RR linking confounder to outcome.
`rr_eu`	RR linking treatment to confounder.
`prevalence_confounder`	Confounder prevalence. Default 0.5.

Value

Named list with `adjusted_estimate`, `bias`, `adjusted_ci_lower`, `adjusted_ci_upper`, `original_estimate`.

`block_bootstrap` *Block bootstrap for dependent / time-series data*

Description

Resamples blocks of consecutive observations. Delegates to `boot::tsboot` when **boot** is installed (fixed and stationary / geometric blocks); a circular-block path is implemented inline because `boot::tsboot` does not expose a circular-block sim mode directly. Falls back to a pure-R loop otherwise.

Usage

```
block_bootstrap(
  data,
  statistic,
  block_size,
  n_boot = 2000L,
  ci_level = 0.95,
  method = "circular",
  seed = 42L
)
```

Arguments

`data`	Numeric vector or matrix.
`statistic`	Function returning a scalar.
`block_size`	Integer block length.
`n_boot`	Number of replicates.
`ci_level`	Confidence level.
`method`	One of `"moving"`, `"circular"`, `"stationary"`.
`seed`	Random seed.

Value

A `morie_bootstrap_result`.

See Also

`boot::tsboot`.

bonferroni *Bonferroni FWER correction*

Description

Thin wrapper over `stats::p.adjust(method = "bonferroni")`.

Usage

```
bonferroni(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

Value

A `morie_rich_result` list (see `morie_multiple_testing`).

bootstrap *Nonparametric bootstrap inference*

Description

Resamples observations with replacement and computes a confidence interval via the percentile, normal, basic, BCa, or studentized method. Delegates the resampling loop and CI computation to `boot::boot` and `boot::boot.ci` when the **boot** package is installed. Falls back to an inline implementation otherwise so the wrapper keeps working on minimal installs. Stratified and cluster resamples are supported in both arms.

Usage

```
bootstrap(
  data,
  statistic,
  n_boot = 2000L,
  ci_level = 0.95,
  ci_method = "bca",
  seed = 42L,
  stratify = NULL,
  cluster = NULL
)
```

Arguments

`data`	A numeric vector or matrix of observations.
`statistic`	Function of one argument that returns a scalar.
`n_boot`	Number of bootstrap replicates (default 2000).
`ci_level`	Confidence level (default 0.95).
`ci_method`	One of `"percentile"`, `"normal"`, `"basic"`, `"bca"`, `"studentized"`.
`seed`	Random seed.
`stratify`	Optional vector of stratum labels (length n).
`cluster`	Optional vector of cluster labels (length n).

Value

A `morie_bootstrap_result` list.

See Also

`boot::boot`, `boot::boot.ci`, `morie_boot_run()`, `morie_boot_basic_ci()`.

`bootstrap_632` *.632 and .632+ bootstrap estimators for prediction error*

Description

Computes apparent error, mean OOB bootstrap error, the .632 estimator (Efron 1983), and the .632+ no-information-adjusted estimator (Efron and Tibshirani 1997). `ipred::errorest(...,estimator = "632plus")` implements the same family in **ipred**; it is cross-referenced for users who already work with **ipred**'s `predict.\<learner\>` ecosystem. The inline implementation is retained because rmorie's API takes naked `model_fn` / `score_fn` callables and is consumed by downstream MRM code.

Usage

```
bootstrap_632(X, y, model_fn, score_fn, n_boot = 200L, seed = 42L)
```

Arguments

`X`	Numeric design matrix (n x p).
`y`	Numeric response (length n).
`model_fn`	Function `model_fn(X_train, y_train)` returning a model object that supports `predict(model, X_test)`.
`score_fn`	Function `score_fn(y_true, y_pred) -> scalar`.
`n_boot`	Number of bootstrap replicates.
`seed`	Random seed.

Value

Named numeric list with apparent_error, bootstrap_error, error_632, error_632plus.

See Also

`ipred::errorest`.

`bootstrap_effect_size_ci` *Generic bootstrap CI wrapper for any effect-size function*

Description

Generic bootstrap CI wrapper for any effect-size function

Usage

```
bootstrap_effect_size_ci(
  func,
  ...,
  n_boot = 2000L,
  confidence = 0.95,
  seed = 42L
)
```

Arguments

`func`	A function taking one or more numeric vectors, returning a scalar.
`...`	Numeric vectors to bootstrap, in `func`'s argument order.
`n_boot`	Bootstrap replicates (default 2000).
`confidence`	Confidence level. Default 0.95.
`seed`	RNG seed. Default 42.

Value

A `morie_effect_size`.

bootstrap_validate *Bootstrap .632 / .632+ validation*

Description

Bootstrap .632 / .632+ validation

Usage

```
bootstrap_validate(
  fit_fn,
  predict_fn,
  X,
  y,
  n_bootstraps = 200L,
  scoring = "roc_auc",
  method = "632plus",
  random_state = 42L
)
```

Arguments

`fit_fn`	A function `(X, y) -> model`.
`predict_fn`	A function `(model, X) -> probability vector`.
`X`	Matrix or data frame of features.
`y`	Vector of targets.
`n_bootstraps`	Number of bootstrap replicates.
`scoring`	”roc_auc”, ”accuracy”, ”brier”.
`method`	”632” or ”632plus”.
`random_state`	Seed.

buttbp *Butterworth bandpass filter*

Description

Zero-phase Butterworth bandpass filter. Isolates a frequency band of interest (e.g., 0.5–40 Hz for EEG, 25–400 Hz for phonocardiogram).

Usage

```
buttbp(x, fs, low, high, order = 4L)
```

Arguments

x	Numeric vector.
fs	Sampling frequency (Hz).
low	Lower cutoff (Hz).
high	Upper cutoff (Hz).
order	Filter order (default 4).

Value

List with `filtered` (numeric vector), `fs`, `order`, `name`.

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  t <- seq(0, 1, length.out = 1000)
  # 2 Hz drift + 10 Hz band of interest + 60 Hz noise
  x <- sin(2 * pi * 2 * t) + sin(2 * pi * 10 * t) +
    0.3 * sin(2 * pi * 60 * t)
  y <- buttbp(x, fs = 1000, low = 5, high = 20)
  length(y$filtered)
}
```

---

buttbs *Butterworth bandstop (notch) filter*

Description

Zero-phase Butterworth bandstop filter. Default 59–61 Hz removes North- American AC mains hum (60 Hz); use 49–51 Hz for European mains.

Usage

```
buttbs(x, fs, low = 59, high = 61, order = 4L)
```

Arguments

x	Numeric vector.
fs	Sampling frequency (Hz).
low	Lower cutoff (Hz, default 59).
high	Upper cutoff (Hz, default 61).
order	Filter order (default 4).

Value

List with `filtered` (numeric vector), `fs`, `order`, `name`.

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  t <- seq(0, 1, length.out = 1000)
  x <- sin(2 * pi * 10 * t) + sin(2 * pi * 60 * t)
  y <- buttbs(x, fs = 1000) # remove 60 Hz mains
  length(y$filtered)
}
```

---

butthp *Butterworth highpass filter*

Description

Zero-phase Butterworth highpass filter. Removes low-frequency drift while preserving higher-frequency content; useful for de-trending physiological signals (EEG, ECG) prior to analysis.

Usage

```
butthp(x, fs, cutoff, order = 4L)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz).
`cutoff`	Cutoff frequency (Hz).
`order`	Filter order (default 4).

Value

List with `filtered` (numeric vector), `fs`, `order`, `name`.

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  t <- seq(0, 1, length.out = 500)
  x <- 5 * t + sin(2 * pi * 10 * t) # linear drift + 10 Hz signal
  y <- butthp(x, fs = 500, cutoff = 1)
  length(y$filtered)
}
```

buttlp *Butterworth lowpass filter*

Description

Zero-phase Butterworth lowpass filter via the suggested `signal` package's `butter()` + `filtfilt()`. Useful for removing high-frequency noise from biological or geophysical time series.

Usage

```
buttlp(x, fs, cutoff, order = 4L)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz).
`cutoff`	Cutoff frequency (Hz).
`order`	Filter order (default 4).

Value

List with `filtered` (numeric vector), `fs`, `order`, `name`.

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  t <- seq(0, 1, length.out = 500)
  x <- sin(2 * pi * 5 * t) + 0.5 * sin(2 * pi * 60 * t) # 5 Hz + 60 Hz
  y <- buttlp(x, fs = 500, cutoff = 20)
  length(y$filtered) # 500
}
```

cauchy_combination *Cauchy combination test (Liu and Xie 2020)*

Description

Robust to arbitrary correlation structure. No clean CRAN equivalent (the ACAT package is GitHub-only); kept as an in-house implementation.

Usage

```
cauchy_combination(p_values, weights = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`weights`	Optional non-negative weights summing to 1.

`CCRSO_TABLE1_RELEASES`

CCRSO Table 1 — 5-year average annual conditional releases

Description

CCRSO Table 1 — 5-year average annual conditional releases

Usage

```
CCRSO_TABLE1_RELEASES
```

Format

An object of class `data.frame` with 3 rows and 10 columns.

`CCRSO_TABLE2_FLOW` *CCRSO Table 2 — prisoner flow 2013/14-2017/18*

Description

CCRSO Table 2 — prisoner flow 2013/14-2017/18

Usage

```
CCRSO_TABLE2_FLOW
```

Format

An object of class `data.frame` with 5 rows and 6 columns.

CCRSO_TABLE3_AGE *CCRSO/StatsCan Table 3 — 2018 age distribution*

Description

CCRSO/StatsCan Table 3 — 2018 age distribution

Usage

```
CCRSO_TABLE3_AGE
```

Format

An object of class `data.frame` with 3 rows and 7 columns.

cepst *Real cepstrum*

Description

Real cepstrum $c[n] = \text{IFFT}(\log|\text{FFT}(x)|)$. Useful for pitch-period estimation and any analysis where the multiplicative magnitude structure of the spectrum is best handled additively in the quefrency domain.

Usage

```
cepst(x, n_fft = NULL)
```

Arguments

`x`	Numeric vector (1-D signal).
`n_fft`	FFT length (default: next power of 2 $\geq$ `length(x)`).

Details

Reference: Rangayyan, R.M. (2015) *Biomedical Signal Analysis*, 2nd ed., Wiley/IEEE Press, chapter on cepstral analysis.

Value

List with `filtered` (real cepstral coefficients), `name`, `fs`, `n_samples`, and `extra` (`quefrency`, `n_fft`).

Examples

```
set.seed(1)
x <- sin(2 * pi * 5 * seq(0, 1, length.out = 512))
res <- cepst(x)
length(res$filtered)
```

`cheatsheet` *Print the morie cheat sheet*

Description

Mirrors the `morie cheatsheet` CLI subcommand: a one-screen reference of install / learn / run / pull / ingest / help commands.

Usage

```
cheatsheet()
```

Value

Invisibly returns a character scalar of the cheatsheet. Called for its side effect of printing to the console.

Examples

```
cheatsheet()
```

`check_referential_integrity` *Check referential integrity (child FK -> parent PK)*

Description

Check referential integrity (child FK -> parent PK)

Usage

```
check_referential_integrity(child, parent, child_key, parent_key)
```

Arguments

`child`	Data frame with foreign key.
`parent`	Data frame with primary key.
`child_key`, `parent_key`	Column names.

chi2_goodness_of_fit *Chi-squared goodness-of-fit test*

Description

Chi-squared goodness-of-fit test

Usage

```
chi2_goodness_of_fit(observed, expected = NULL)
```

Arguments

observed	Observed counts.
expected	Expected counts or NULL for uniform.

chi2_independence *Chi-squared test of independence*

Description

Chi-squared test of independence

Usage

```
chi2_independence(contingency_table, correction = TRUE)
```

Arguments

contingency_table	A matrix or table of counts.
correction	Yates's continuity correction (2x2).

ckan_metadata *CKAN Metadata for Open Data APIs*

Description

Package IDs and metadata URLs for accessing CPADS, CSADS, and CSUS datasets via the Canadian Open Data CKAN API.

Usage

```
ckan_metadata
```

Format

A data.frame with columns:

survey Survey abbreviation: cpads, csads, csus

name Full survey name

package_id CKAN package UUID

metadata_url URL to retrieve full package metadata

Source

https://open.canada.ca

Examples

```
data(ckan_metadata)
ckan_metadata$metadata_url
```

cles *Common Language Effect Size (probability of superiority)*

Description

Estimates P(X > Y) for randomly drawn observations from each group.

Usage

```
cles(x, y, confidence = 0.95)
```

Arguments

x, y	Numeric vectors (NA dropped).
confidence	Confidence level for CI. Default 0.95.

Value

A `morie_effect_size`.

---

cliffs_delta *Cliff's delta*

Description

Cliff's delta

Usage

```
cliffs_delta(x, y, confidence = 0.95)
```

Arguments

`x`, `y`	Numeric vectors (NA dropped).
`confidence`	Confidence level for CI. Default 0.95.

Value

A `morie_effect_size`.

---

cochrans_q *Cochran's Q test*

Description

Cochran's Q test

Usage

```
cochrans_q(...)
```

Arguments

`...`	Three or more matched binary 0/1 vectors.

coef.crr *S3 coef method for cmprsk::crr objects.*

Description

cmprsk::crr does not ship a `coef.crr` method, so a bare `stats::coef()` on a `crr` fit falls through to `coef.default()` and returns `NULL`. morie registers this method (via S3method in NAMESPACE, generated by roxygen `@exportS3Method`) so the standard accessor returns the fitted coefficient vector for any caller – not just `morie_survival_finegray`.

Usage

```
## S3 method for class 'crr'
coef(object, ...)
```

Arguments

`object`	A `cmprsk::crr` fit.
`...`	Ignored.

Value

Named numeric vector of regression coefficients.

coefficient_of_variation *Coefficient of variation*

Description

Coefficient of variation

Usage

```
coefficient_of_variation(x)
```

Arguments

`x`	Numeric vector.

Value

A `morie_effect_size`.

cohens_f — *Cohen's f from eta-squared*

Description

Cohen's f from eta-squared

Usage

```
cohens_f(eta2)
```

Arguments

`eta2`	Eta-squared value.

Value

A `morie_effect_size`.

cohens_kappa — *Cohen's kappa for two raters*

Description

Cohen's kappa for two raters

Usage

```
cohens_kappa(rater1, rater2, confidence = 0.95)
```

Arguments

`rater1`, `rater2`	Equal-length categorical vectors.
`confidence`	Confidence level.

cohens_w *Cohen's w for chi-squared*

Description

Cohen's w for chi-squared

Usage

```
cohens_w(observed, expected = NULL)
```

Arguments

observed	Observed frequencies (numeric vector).
expected	Expected frequencies (or NULL for uniform).

Value

A morie_effect_size.

collinearity_diagnostics *Comprehensive multicollinearity diagnostics*

Description

Comprehensive multicollinearity diagnostics

Usage

```
collinearity_diagnostics(X, column_names = NULL)
```

Arguments

X	Design matrix.
column_names	Optional column names.

Value

A morie_collinearity_diagnostics list.

`column_rule` *Construct a column rule*

Description

Construct a column rule

Usage

```
column_rule(
  name,
  dtype = NULL,
  required = TRUE,
  nullable = TRUE,
  null_threshold = 1,
  min_val = NULL,
  max_val = NULL,
  allowed_values = NULL,
  unique = FALSE,
  regex = NULL,
  custom = NULL
)
```

Arguments

`name`	Column name.
`dtype`	One of "numeric", "character"/"object", "datetime", or NULL.
`required`	Whether the column must be present.
`nullable`	Whether NA values are allowed.
`null_threshold`	Maximum allowed fraction of NA (0–1).
`min_val`, `max_val`	Numeric bounds (or NULL).
`allowed_values`	Vector of permitted values (or NULL).
`unique`	Logical; whether values must be unique.
`regex`	Regex pattern for string columns (or NULL).
`custom`	Optional function `(column) -> logical(1)`.

Value

A `column_rule` list.

commands_by_category *Commands grouped by category*

Description

Commands grouped by category

Usage

```
commands_by_category()
```

Value

Named list of character vectors of command names.

compute_goodness_of_fit *Comprehensive goodness-of-fit statistics*

Description

R^2 and adjusted R^2 for linear models; McFadden pseudo-R^2, deviance and Pearson chi-squared for logistic / Poisson; AIC, BIC, log-likelihood, and the omnibus F-test for linear models.

Usage

```
compute_goodness_of_fit(
  y,
  y_hat,
  X,
  model_type = "linear",
  log_likelihood = NULL
)
```

Arguments

`y`	Response vector.
`y_hat`	Fitted values.
`X`	Design matrix.
`model_type`	`"linear"`, `"logistic"`, `"poisson"`.
`log_likelihood`	Optional precomputed log-likelihood.

Value

A `morie_goodness_of_fit` list.

compute_influence *Compute leverage and influence diagnostics*

Description

Hat-matrix diagonal, Cook's distance (`stats::cooks.distance` is preferred for fitted `lm`s; this function works straight from `X`, `y`, and a fitted `y_hat`), DFFITS, DFBETAS, and COVRATIO.

Usage

```
compute_influence(y, X, y_hat = NULL)
```

Arguments

`y`	Response vector.
`X`	Design matrix.
`y_hat`	Optional fitted values (OLS is used if NULL).

Value

A `morie_influence_diagnostics` list.

compute_residuals *Compute residual diagnostics*

Description

Returns raw / standardised / externally studentised residuals along with normality, heteroskedasticity (Breusch-Pagan), and autocorrelation (Durbin-Watson) tests. Optionally also returns deviance and Pearson residuals for logistic / Poisson GLMs.

Usage

```
compute_residuals(y, y_hat, X, model_type = "linear")
```

Arguments

`y`	Observed response.
`y_hat`	Fitted values.
`X`	Design matrix.
`model_type`	`"linear"`, `"logistic"`, or `"poisson"`.

Value

A `morie_residual_diagnostics` list.

compute_vif *Variance Inflation Factors*

Description

For each column j of X, regresses column j on the remaining columns (plus an intercept) and returns 1/(1 - R^2).

Usage

```
compute_vif(X, column_names = NULL)
```

Arguments

X	Design matrix (without intercept).
column_names	Optional character vector of names.

Value

Named numeric vector of VIFs.

correlation_matrix *Pairwise correlation matrix with p-values*

Description

Pairwise correlation matrix with p-values

Usage

```
correlation_matrix(data, method = "pearson")
```

Arguments

data	Data frame; numeric columns are used.
method	One of "pearson", "spearman", "kendall".

Value

List with components `r` (correlations) and `p` (p-values), both `data.frame` objects with matching dimensions.

correlation_table *Pairwise correlation matrix with significance stars*

Description

Pairwise correlation matrix with significance stars

Usage

```
correlation_table(
  data,
  method = "pearson",
  show_stars = TRUE,
  mask_diagonal = TRUE,
  digits = 3L,
  output_format = "dataframe",
  title = "Correlation Matrix"
)
```

Arguments

data	Data frame.
method	”pearson”, ”spearman”, ”kendall”.
show_stars	Annotate cells with significance stars.
mask_diagonal	Replace diagonal with ”-”.
digits	Decimal places.
output_format	Output target.
title	Title.

create_reproducibility_manifest *Create a manifest of the environment for reproducibility*

Description

Create a manifest of the environment for reproducibility

Usage

```
create_reproducibility_manifest(data, parameters = NULL, seeds = NULL)
```

Arguments

data	Data frame (used for a SHA-256 checksum).
parameters	Optional list of analysis parameters.
seeds	Optional named list of random seeds.

`cross_validate` *Cross-validate a model using a user-supplied fit/predict pair*

Description

Cross-validate a model using a user-supplied fit/predict pair

Usage

```
cross_validate(
  fit_fn,
  predict_fn,
  X,
  y,
  method = "stratified_kfold",
  n_folds = 5L,
  n_repeats = 10L,
  scoring = "roc_auc",
  groups = NULL,
  confidence = 0.95,
  random_state = 42L
)
```

Arguments

`fit_fn`	A function `(X, y) -> model`.
`predict_fn`	A function `(model, X) -> probability vector`.
`X`	Matrix or data frame of features.
`y`	Vector of targets.
`method`	Resampling strategy: "kfold", "stratified_kfold", "grouped_kfold", "loo", "monte_carlo", "time_series".
`n_folds`	Number of folds.
`n_repeats`	Repeats for monte_carlo.
`scoring`	"roc_auc", "accuracy", "brier".
`groups`	Group labels for grouped_kfold.
`confidence`	Confidence level for the score CI.
`random_state`	Seed.

d_to_nnt — *Convert Cohen's d to NNT given a control event rate*

Description

Uses the Kraemer & Kupfer (2006) approximation.

Usage

```
d_to_nnt(d, base_rate = 0.5)
```

Arguments

d	Cohen's d.
base_rate	Control event rate (default 0.5).

Value

Numeric NNT.

d_to_or — *Convert Cohen's d to odds ratio*

Description

Convert Cohen's d to odds ratio

Usage

```
d_to_or(d)
```

Arguments

d	Cohen's d.

Value

Numeric OR.

d_to_r *Convert Cohen's d to Pearson r*

Description

Convert Cohen's d to Pearson r

Usage

```
d_to_r(d, n1 = NULL, n2 = NULL)
```

Arguments

d	Cohen's d.
n1, n2	Sample sizes (or NULL).

Value

Numeric r.

dagostino_pearson *D'Agostino-Pearson omnibus normality test*

Description

API stub: implemented via the K2 statistic = Z(skew)^2 + Z(kurt)^2 (D'Agostino & Pearson 1973). Recommended n >= 20.

Usage

```
dagostino_pearson(x)
```

Arguments

x	Numeric vector.

`dataset_catalog` *MORIE Dataset Catalog*

Description

A data.frame listing all Canadian public health datasets available through the MORIE data management system. Each row describes one dataset with its source, survey, year, format, and access metadata.

Usage

```
dataset_catalog
```

Format

A data.frame with columns:

key Unique catalog key (e.g., `"opencanada_cpads_2021"`)

name Human-readable dataset name

source Data source: opencanada, healthinfobase, or cihi

survey Survey abbreviation: cpads, ccs, csads, csus, or indicators

year Year or year range (e.g., `"2021-2022"`)

format File format: csv or xlsx

type Data type: pumf, bootstrap, aggregate, or indicator

large_file Logical; TRUE for bootstrap weight files (>100MB)

local_path Relative path to the local data file

table_name SQLite table name in the DBI cache

ckan_resource_id CKAN DataStore resource ID (empty if unavailable)

Source

Health Canada, CIHI, Statistics Canada open data portals.

Examples

```
data(dataset_catalog)
head(dataset_catalog)
```

`decision_curve_analysis` *Decision curve analysis*

Description

Decision curve analysis

Usage

```
decision_curve_analysis(y_true, y_pred, thresholds = NULL)
```

Arguments

`y_true`	Integer 0/1 vector.
`y_pred`	Predicted probabilities.
`thresholds`	Numeric vector of thresholds (defaults to `seq(0.01, 0.99, 0.01)`).

`decoupling_test` *Doob decoupling test*

Description

Tests Doob's central thesis that imprisonment is decoupled from crime by computing the Pearson correlation between the two time series and (optionally) running Pettitt change-point on each.

Usage

```
decoupling_test(crime_series, imprisonment_series, years = NULL)
```

Arguments

`crime_series`	Numeric vector of per-period crime rates.
`imprisonment_series`	Numeric vector of per-period imprisonment rates, same length as `crime_series`.
`years`	Optional integer vector of period labels.

Value

A `morie_result` named-list.

`delete_d_jackknife` *Delete-d (generalised) jackknife*

Description

Generalised jackknife removing `d` observations per replicate; all subsets are enumerated when $C(n, d) \leq$ `max_subsets` and Monte-Carlo sampled otherwise. **resample** exposes the equivalent generalised jackknife API via `resample::jackknife` (cross-referenced).

Usage

```
delete_d_jackknife(
  data,
  statistic,
  d = 2L,
  ci_level = 0.95,
  max_subsets = 5000L,
  seed = 42L
)
```

Arguments

`data`	Numeric vector or matrix.
`statistic`	Function returning a scalar.
`d`	Number of observations to delete per replicate.
`ci_level`	Confidence level.
`max_subsets`	Maximum subsets to evaluate.
`seed`	Random seed.

Value

A `morie_jackknife_result`.

See Also

`resample::jackknife`.

detect_overfitting *Bootstrap optimism-corrected performance*

Description

Bootstrap optimism-corrected performance

Usage

```
detect_overfitting(
  fit_fn,
  predict_fn,
  X,
  y,
  scoring = "roc_auc",
  n_bootstrap = 200L,
  random_state = 42L
)
```

Arguments

fit_fn	A function (X, y) -> model.
predict_fn	A function (model, X) -> probability vector.
X	Matrix or data frame of features.
y	Vector of targets.
scoring	"roc_auc", "accuracy", "brier".
n_bootstrap	Integer; number of bootstrap resamples used to estimate the optimism correction (default 200).
random_state	Seed.

dfa *Detrended fluctuation analysis (DFA)*

Description

Estimates the DFA scaling exponent α. White noise gives $\alpha \approx 0.5$; pink (1/f) noise $\alpha \approx 1.0$; Brownian motion $\alpha \approx 1.5$.

Usage

```
dfa(x, scales = NULL)
```

Arguments

`x`	Numeric vector (length ≥ 16).
`scales`	Integer vector of window sizes (auto-generated if `NULL`).

Details

Reference: Peng, C.-K., Havlin, S., Stanley, H.E. & Goldberger, A.L. (1995) "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series", *Chaos* 5(1):82–87.

Value

List with `value` (alpha), `name`, and `extra` (`scales`, `fluctuation`).

Examples

```
set.seed(1)
x <- cumsum(rnorm(2048))
res <- dfa(x)
res$value
```

`doob_trends` *Doob Federal Court Affidavit — national-aggregate trend analyses*

Description

Replicates the analytical contribution of Prof. Anthony N. Doob's expert-witness affidavit in *Canadian Civil Liberties Association et al. v. The Attorney General of Canada* (Federal Court file T-539-20, Application Record Vol. 3 of 5, pp. 778-795).

Details

Doob's national-aggregate analyses (Figures 1-4 + Tables 1-3) sit ALONGSIDE the per-row MRM modules on OTIS provincial data and the MRM chi-square family on aggregate contingency tables.

References

Doob, A. N. (2020). Affidavit (T-539-20) of Anthony Doob — Federal Court of Canada, Application Record Vol. 3 of 5. CCLA et al. v. Attorney General of Canada.

e_value — *E-value for unmeasured confounding (continuous-ATE scale)*

Description

Thin wrapper over `EValue::evalues.OLS()` when **EValue** is installed and an outcome SD is supplied via the `sd_y` argument (recommended workflow per VanderWeele & Ding 2017). Without **EValue**, or when `sd_y` is left at its default of 1, falls back to the closed-form continuous-scale RR proxy used by the Python port so both ports stay numerically aligned.

Usage

```
e_value(ate, se, null = 0, sd_y = 1)
```

Arguments

`ate`	Point estimate of the treatment effect.
`se`	Standard error of the ATE (must be > 0).
`null`	Null value. Default 0.
`sd_y`	Outcome standard deviation. Default 1 (use the closed-form proxy). Pass the empirical sd to route through `EValue::evalues.OLS()` when installed.

Value

Scalar E-value ($>= 1$).

e_value_d — *E-value for a standardised mean difference (Cohen's d)*

Description

Converts d to an RR scale via the VanderWeele-Ding approximation RR ~ exp(0.91 * d), then applies `e_value_rr()`.

Usage

```
e_value_d(d, se = NULL, n = NULL)
```

Arguments

`d`	Standardised mean difference.
`se`	Standard error of d (optional).
`n`	Sample size for SE approximation (optional).

Value

A `morie_evalue` named-list.

---

e_value_hr *E-value for a hazard ratio*

Description

Uses the HR-to-RR approximation from VanderWeele (2017).

Usage

```
e_value_hr(hr, ci_lower = NULL, ci_upper = NULL)
```

Arguments

`hr`	Hazard ratio.
`ci_lower`, `ci_upper`	Optional 95% CI of HR.

Value

A `morie_evalue` named-list.

---

e_value_or *E-value for an odds ratio*

Description

Uses Zhang & Yu (1998) OR-to-RR correction when `prevalence >= 0.15`.

Usage

```
e_value_or(odds_ratio, ci_lower = NULL, ci_upper = NULL, prevalence = NULL)
```

Arguments

`odds_ratio`	Observed odds ratio.
`ci_lower`, `ci_upper`	Optional 95% CI.
`prevalence`	Outcome prevalence (optional).

Value

A `morie_evalue` named-list.

e_value_rr *E-value for a risk ratio*

Description

Wraps **EValue** when available; otherwise applies the VanderWeele-Ding closed-form formula directly.

Usage

```
e_value_rr(rr, ci_lower = NULL, ci_upper = NULL)
```

Arguments

`rr`	Observed risk ratio.
`ci_lower`	Lower 95% CI of the RR (optional).
`ci_upper`	Upper 95% CI of the RR (optional).

Value

A `morie_evalue` named-list.

ecgdet *Pan-Tompkins QRS / R-peak detector*

Description

Pan-Tompkins QRS detection: bandpass (5–15 Hz) -> differentiate -> square -> moving-window integration -> adaptive thresholding -> refinement against the raw ECG.

Usage

```
ecgdet(ecg, fs)
```

Arguments

`ecg`	Numeric vector (1-D ECG signal).
`fs`	Sampling frequency in Hz.

Details

Reference: Pan, J. & Tompkins, W.J. (1985) "A real-time QRS detection algorithm", *IEEE Trans. Biomed. Eng.* BME-32(3):230–236.

Value

List with `filtered` (raw ECG echoed), `name`, `fs`, `n_samples`, and `extra` (`r_peaks` = 1-based sample indices, `n_peaks`).

Examples

```
set.seed(1)
fs <- 250
t <- seq(0, 4, by = 1 / fs)
ecg <- sin(2 * pi * 1.2 * t) + 0.1 * rnorm(length(t))
if (requireNamespace("signal", quietly = TRUE)) {
  res <- ecgdet(ecg, fs)
  res$extra$n_peaks
}
```

`effect_size_result` *Build an effect-size result*

Description

Returns a named-list with class `c("morie_effect_size", "list")`.

Usage

```
effect_size_result(
  measure,
  estimate,
  ci_lower = NA_real_,
  ci_upper = NA_real_,
  se = NA_real_,
  n = NA_integer_,
  extra = list()
)
```

Arguments

`measure`	Name of the effect-size statistic.
`estimate`	Point estimate (numeric).
`ci_lower`	Lower confidence bound (or NA).
`ci_upper`	Upper confidence bound (or NA).
`se`	Standard error (or NA).
`n`	Sample size (or NA).
`extra`	Named list of additional outputs.

Value

A `morie_effect_size` named-list.

`effect_sizes` *Comprehensive effect-size calculations*

Description

Effect-size estimators used in biomedical and social-science research, each with analytic or bootstrap confidence intervals.

Details

Families: standardised mean differences (Cohen's d, Hedges' g, Glass's delta); common-language ES (CLES); correlation-based (r, R^2, eta^2, partial eta^2, omega^2, epsilon^2); contingency (OR, RR, RD, NNT, NNH, rate ratio, IRD); association (Cohen's w, Cramer's V, phi); non-parametric (rank-biserial, Cliff's delta, Vargha-Delaney A); regression (standardised beta, CV); and meta- analysis (fixed-/random-effects pooling, I^2, prediction interval).

References

Cohen (1988); Hedges & Olkin (1985); Borenstein et al. (2009); Vargha & Delaney (2000); DerSimonian & Laird (1986).

`effects` *Treatment effect estimators and marginal-effects extenders*

Description

Two function families live here:

Details

Treatment-effect estimators (legacy)

- `estimate_ate()` – IPW-weighted OLS ATE.
- `estimate_plr()` – Partially Linear Regression via **DoubleML**; falls back to base R cross-fit ridge.
- `estimate_pliv()` – Partially Linear IV (LATE) via **DoubleML**; falls back to 2SLS.
- `estimate_ate_gcomputation()` – G-computation ATE. Thin wrapper over `stdReg::stdGlm()` when installed; falls back to inline bootstrap standardisation otherwise.
- `sensitivity_rosenbaum()` – Rosenbaum bounds. Thin wrapper over `rbounds::psens()` when installed; otherwise normal-approximation Wilcoxon signed-rank bounds.

- `e_value()` – VanderWeele-Ding E-value. Thin wrapper over `EValue::evalues.OLS()` when installed; otherwise the closed-form continuous-scale RR proxy.

Marginal-effects extenders (Phase 1.j additions; thin wrappers over Vincent Arel-Bundock's universal API and the **emmeans** / **broom** ecosystems):

- `morie_effects_emmeans()` -> `emmeans::emmeans()`.
- `morie_effects_predictions()` -> `marginaleffects::predictions()`.
- `morie_effects_comparisons()` -> `marginaleffects::comparisons()`.
- `morie_effects_slopes()` -> `marginaleffects::slopes()`.
- `morie_effects_tidy()` -> `broom::tidy()` (falls back to a `summary()`-based tidy frame when **broom** is unavailable).

Each extender requires the underlying CRAN package and signals a clean `stop()` when it is missing, leaving the upstream model object untouched. They return the underlying package's native object verbatim so downstream code (e.g. **ggplot2** plumbing, rmorie's MRM step) keeps working with the canonical API.

References

Chernozhukov et al. (2018); Robins (1986); VanderWeele & Ding (2017); Rosenbaum (2002); Arel-Bundock (2024, **marginaleffects**); Lenth (2024, **emmeans**); Robinson, Hayes & Couch (2024, **broom**).

`epsilon_squared` *Epsilon-squared (Kelley, 1935)*

Description

Epsilon-squared (Kelley, 1935)

Usage

```
epsilon_squared(ss_effect, ss_total, df_effect, ms_error)
```

Arguments

`ss_effect`, `ss_total`	Sums of squares.
`df_effect`	Numerator d.f. of the effect.
`ms_error`	Error mean square.

Value

A `morie_effect_size`.

`estimate_ate` *IPW-weighted OLS ATE*

Description

Thin wrapper over a weighted `stats::lm()` plus HC3 robust SEs from **sandwich** + **lmtest** when installed. Note: this is the legacy shape used by older MRM pipelines; new code should prefer `morie_estimate_ate()` (in `R/causal.R`) for the richer `morie_te_result` return shape.

Usage

```
estimate_ate(data, outcome, treatment, weights_col)
```

Arguments

`data`	Data frame containing the analytical sample.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary treatment column.
`weights_col`	Name of the weights column (e.g. IPTW).

Value

Named list with `ate` and `se` (HC3-robust when available).

`estimate_ate_gcomputation`
G-computation ATE with bootstrap SE

Description

Thin wrapper over `stdReg::stdGlm()` (Sjolander's regression-standardisation back end) when **stdReg** is installed. Without **stdReg**, falls back to an inline outcome- regression + bootstrap implementation (500 resamples, seed 42) that mirrors the legacy rmorie behaviour.

Usage

```
estimate_ate_gcomputation(
  data,
  treatment,
  outcome,
  covariates,
  outcome_model = "linear"
)
```

Arguments

`data`	Data frame with all required columns.
`treatment`	Binary treatment column (0/1).
`outcome`	Outcome column.
`covariates`	Character vector of covariates.
`outcome_model`	`"linear"` (OLS) or `"logistic"` (logit GLM).

Value

Named list with `ate`, `se`, `ci_lower`, `ci_upper`, `n_obs`, `outcome_model`.

`estimate_pi0` *Estimate the proportion of true null hypotheses (pi0)*

Description

Delegates to `qvalue::pi0est` (Bioconductor) when installed and `method = "storey"`. Otherwise computes the cutoff-based or bootstrap-based estimator inline.

Usage

```
estimate_pi0(p_values, method = c("storey", "bootstrap", "two_step"))
```

Arguments

`p_values`	Numeric vector of raw p-values.
`method`	One of `"storey"`, `"bootstrap"`, or `"two_step"`.

Value

A scalar pi0 estimate in `[0, 1]`.

`estimate_pliv` *Partially Linear IV (PLIV) / Local Average Treatment Effect*

Description

Wraps **DoubleML** when available. Otherwise falls back to 2SLS: first stage `D ~ Z + X`, second stage `Y ~ D_hat + X`, base R OLS.

Usage

```
estimate_pliv(
  data,
  treatment,
  outcome,
  instrument,
  covariates,
  n_folds = 5L,
  random_state = 42L
)
```

Arguments

`data`	Data frame with all required columns.
`treatment`	Endogenous treatment column name.
`outcome`	Outcome column name.
`instrument`	Instrument column name.
`covariates`	Exogenous covariate column names.
`n_folds`	Cross-fitting folds (DoubleML path). Default 5.
`random_state`	RNG seed. Default 42.

Value

Named list with `late`, `se`, `ci_lower`, `ci_upper`, `pval`, `n_obs`, `method`.

`estimate_plr` *Partially Linear Regression (PLR) ATE*

Description

Wraps **DoubleML** (+ **mlr3** / **mlr3learners**) when available. Without DoubleML, falls back to a hand-rolled cross-fitting estimator using ridge regression (**glmnet**) or, last-ditch, OLS partialling out.

Usage

```
estimate_plr(
  data,
  treatment,
  outcome,
  covariates,
  n_folds = 5L,
  random_state = 42L
)
```

Arguments

`data`	Data frame with all required columns.
`treatment`	Column name of the treatment variable.
`outcome`	Column name of the outcome variable.
`covariates`	Character vector of covariate column names.
`n_folds`	Cross-fitting folds. Default 5.
`random_state`	RNG seed. Default 42.

Value

Named list with `ate`, `se`, `ci_lower`, `ci_upper`, `pval`, `n_obs`, `method`.

`explain_file` *Human-readable description of a morie output CSV*

Description

Looks up a one-paragraph + short-table explanation by filename (any leading directory components are stripped). Falls back to matching on the filename stem if the extension differs.

Usage

```
explain_file(filename)
```

Arguments

`filename`	The CSV filename, with or without a path.

Value

A character scalar containing the explanation. If no registered entry matches, returns a fallback listing the known files.

Examples

```
cat(explain_file("power_summary.csv"))
```

explain_known_files *Names of all morie output CSVs with registered explanations*

Description

Names of all morie output CSVs with registered explanations

Usage

```
explain_known_files()
```

Value

Character vector of filenames.

extenders_nonparam *FDR / nonparametric / quantile / latent-class extenders (Phase 1.n)*

Description

Thin wrapper-as-extender entry points that delegate to canonical CRAN packages for local-FDR estimation (**locfdr**), FDR / q-values (**fdrtool**), quantile regression (**quantreg**), nonparametric kernel regression (**np**), Bayesian nonparametric Dirichlet-process mixtures (**dirichletprocess**), and latent-class mixed models (**lcmm**). Each function returns a two-element list with `$method` (the qualified upstream function name, or a multi-step string for compound pipelines) and `$raw` (the upstream return object), so downstream callers can pattern-match on shape while keeping the full upstream object available for inspection.

extenders_rdd *RDD / IRT extenders (Phase 1.l)*

Description

Thin wrapper-as-extender entry points that delegate to canonical CRAN packages for regression-discontinuity diagnostics (**rddensity**, **rdlocrand**, **rdpower**), ordinal vignette analysis (**anchors**), and alpha-NOMINATE ideal-point estimation (**anominate**). Each function returns a two-element list with `$method` (the qualified upstream function name) and `$raw` (the upstream return object), so downstream callers can pattern-match on shape while keeping the full upstream object available for inspection.

extenders_spatial *Spatial / multivariate / meta-analysis extenders (Phase 1.m)*

Description

Thin wrapper-as-extender entry points that delegate to canonical CRAN packages for geostatistics (**gstat**), multivariate dependence (**copula**), kernel methods (**kernlab**), meta-analysis (**metafor**) and the multivariate normal / t distributions (**mvtnorm**). Each function returns a two-element list with `$method` (the qualified upstream function name) and `$raw` (the upstream return object), so downstream callers can pattern-match on shape while keeping the full upstream object available for inspection.

extenders_stats *Stats extenders (Phase 1.k)*

Description

Thin wrapper-as-extender entry points that delegate to canonical CRAN statistics packages. Each function returns a two-element list with `$method` (the qualified upstream function name) and `$raw` (the upstream return object), so downstream callers can pattern-match on shape while keeping the full upstream object available for inspection.

external_validate *External validation on new data*

Description

External validation on new data

Usage

```
external_validate(predict_fn, X_external, y_external, X_development = NULL)
```

Arguments

`predict_fn`	A function `(X) -> probability vector` (model already bound by closure).
`X_external`	Matrix or data frame of features.
`y_external`	Outcome vector.
`X_development`	Optional development-data features for KS-based domain-shift diagnostics.

`fairness_cityprofile` *City-agnostic data profiles for the predictive-policing audit*

Description

R port of the Python module `morie.fairness.cityprofile`. The disparity audit operates on a canonical per-area schema: `area`, `risk`, `outcome`, `population`, `group`. A `morie_city_profile` records which columns of one city's open-data export carry those five canonical fields, and `morie_fairness_apply_profile` renames an arbitrary city `data.frame` onto the canonical schema so the audit code never needs to know which city the data came from.

Details

Functions:

- `morie_fairness_city_profile`: constructor for a city profile object.
- `morie_fairness_register_city`: register a profile in the process-local registry.
- `morie_fairness_get_city`: look up a registered profile by case-insensitive name.
- `morie_fairness_list_cities`: list registered profile names.
- `morie_fairness_apply_profile`: rename a `data.frame` onto the canonical schema.

`fairness_metrics` *Group-disparity metrics for auditing classification systems*

Description

R port of `morie.fairness.metrics`. Each callable is an *audit* measure: given decisions a system made (and, where available, the realised ground truth) plus a protected attribute, it quantifies whether outcomes differ across groups. None of these functions make predictions; they only measure disparity in predictions that already exist.

Details

Functions:

- `morie_fairness_disparate_impact`: the four-fifths rule.
- `morie_fairness_demographic_parity`: favourable-rate gap.
- `morie_fairness_equalized_odds`: TPR/FPR gaps (needs ground truth).
- `morie_fairness_average_odds_difference`: mean TPR+FPR gap.
- `morie_fairness_gini`: concentration of a score distribution.
- `morie_fairness_bias_amplification`: composite of parity gap and inequality.

Prior art reimplemented independently (no code copied): the COMPAS fairness audit in pbiecek's *XAI Stories* and IBM's AI Fairness 360 definitions; the predictive-policing disparity framing of the SciencesPo *Predictive-policing-Chicago* project (Lacherade, Szabo, Krikava & Aeby, 2021) and Barman & Barman, arXiv:2603.18987.

fairness_temporal *Multi-city temporal disparity audit*

Description

R port of `morie.fairness.temporal`. Reimplements Barman & Barman, *Unmasking Algorithmic Bias in Predictive Policing* (arXiv:2603.18987): the four disparity metrics - Disparate Impact Ratio, Demographic Parity Gap, Gini coefficient, and Bias Amplification Score - are computed for every (`city`, `period`) cell and assembled into a time series so that *temporal instability* and *cross-city divergence* become visible.

Details

Builds on the metrics in `fairness_metrics`.

fallback_procedure *Fallback (fixed-sequence with alpha spending)*

Description

Wiens-Dmitrienko fallback; **gMCP** can express the same procedure as a graphical multiple-comparison transition.

Usage

```
fallback_procedure(p_values, weights, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`weights`	Numeric vector of non-negative weights summing to 1.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

fisher_combined — *Fisher's method for combining independent p-values*

Description

Delegates to `poolr::fisher` when installed; otherwise computes the chi-square statistic inline.

Usage

```
fisher_combined(p_values)
```

Arguments

`p_values`	Numeric vector of raw p-values.

fisher_exact_test — *Fisher's exact test for a 2x2 table*

Description

Fisher's exact test for a 2x2 table

Usage

```
fisher_exact_test(contingency_table, alternative = "two.sided")
```

Arguments

`contingency_table`	2x2 matrix.
`alternative`	One of "two.sided", "less", "greater".

fixed_effects_meta *Fixed-effects (inverse-variance) meta-analytic pooling*

Description

Fixed-effects (inverse-variance) meta-analytic pooling

Usage

```
fixed_effects_meta(estimates, standard_errors, confidence = 0.95)
```

Arguments

estimates	Numeric vector of effect-size estimates.
standard_errors	Numeric vector of SEs.
confidence	Confidence level. Default 0.95.

Value

A `morie_effect_size` with Q + Q p-value in `extra`.

fixed_sequence *Fixed-sequence (predetermined order) testing*

Description

Tests are evaluated in the given order and the procedure stops at the first non-rejection; reached hypotheses need no multiplicity adjustment. The **gMCP** package implements the equivalent procedure via a chain graph.

Usage

```
fixed_sequence(p_values, alpha = 0.05, labels = NULL)
```

Arguments

p_values	Numeric vector of raw p-values.
alpha	Significance level.
labels	Optional character vector of test labels.

format_dataframe *Apply uniform formatting to numeric columns*

Description

Apply uniform formatting to numeric columns

Usage

```
format_dataframe(
  df,
  numeric_fmt = "%.2f",
  pval_cols = NULL,
  output_format = "dataframe",
  title = ""
)
```

Arguments

df	Data frame.
numeric_fmt	sprintf-style format spec without leading "%".
pval_cols	Columns to format as p-values.
output_format	Output target.
title	Title.

format_number *Format a single number according to style conventions*

Description

Format a single number according to style conventions

Usage

```
format_number(
  x,
  style = c("fixed", "scientific", "percent", "integer"),
  digits = 2L,
  apa = FALSE
)
```

Arguments

x	Numeric.
style	"fixed", "scientific", "percent", "integer".
digits	Decimal places.
apa	APA-style leading-zero suppression.

---

friedman_test *Friedman test (repeated-measures rank ANOVA)*

Description

Friedman test (repeated-measures rank ANOVA)

Usage

```
friedman_test(...)
```

Arguments

...	Three or more equal-length numeric vectors.

---

frns_metrics *Group-disparity metrics for auditing classification and risk systems*

Description

R parity for the Python `morie.fairness.metrics` module. Every callable here is an *audit* measure: given the decisions a system made (and, where available, the realised ground truth) plus a protected attribute such as race, it quantifies whether outcomes differ across groups. None of these functions make predictions; they only measure disparity in predictions that already exist.

Details

Functions:

- `morie_fairness_disparate_impact()`: the EEOC four-fifths rule.
- `morie_fairness_demographic_parity()`: favourable-rate gap.
- `morie_fairness_equalized_odds()`: TPR/FPR gaps (needs ground truth).
- `morie_fairness_average_odds_difference()`: mean TPR+FPR gap.
- `morie_fairness_gini()`: concentration of a score distribution.
- `morie_fairness_bias_amplification()`: composite `Delta_parity * G`.

Each returns a named `list` with the metric value, a per-group breakdown, any advisory `warnings`, and a plain-language `interpretation`, mirroring the payload of the Python `RichResult`.

Prior art reimplemented independently (no code copied): IBM AI Fairness 360 metric definitions; the COMPAS audit in pbiecek's *XAI Stories*; the SciencesPo *Predictive-policing-Chicago* project (Lacherade, Szabo, Krikava & Aeby, 2021); and Barman & Barman, arXiv:2603.18987 (the Bias Amplification Score).

Value

Each callable in this module returns a named `list` with the metric `value`, a per-group breakdown, advisory `warnings`, and a plain-language `interpretation`.

Examples

```
pred <- c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0)
race <- c(rep("A", 5), rep("B", 5))
morie_fairness_disparate_impact(pred, race, privileged = "A")$value
```

---

frns_predpol *Generalised predictive-policing disparity audit*

Description

R parity for the Python `morie.fairness.predpol` module. A clean-room, city-agnostic reimplementation of the district-level analysis of the SciencesPo *Predictive-policing-Chicago* project (Lacherade, Szabo, Krikava & Aeby, 2021): rank areas by the risk an algorithm predicts, rank them by their realised outcome rate, and test whether the disagreement tracks the areas' demographic composition.

Details

Functions:

- `morie_predpol_aggregate_areas()`: roll per-record data up to one row per area.
- `morie_predpol_calibration_audit()`: Spearman calibration plus a per-group mean rank gap (the over-/under-prediction signal).
- `morie_predpol_score_disparity()`: descriptive per-group risk-score summary with a one-way ANOVA.

Written from the project's published methodology; no code copied (that repository carries no licence and is not redistributable).

Value

`morie_predpol_aggregate_areas()` returns a per-area `data.frame`; `morie_predpol_calibration_audit()` and `morie_predpol_score_disparity()` return named `list`s of audit statistics, per-group breakdowns, and a plain-language `interpretation`.

Examples

```
agg <- morie_predpol_aggregate_areas(
  area = c("a", "a", "b", "b"), risk = c(10, 20, 30, 40),
  outcome = c(1, 0, 1, 1)
)
agg$mean_risk
```

---

frns_temporal *Multi-city temporal disparity audit*

Description

R parity for the Python `morie.fairness.temporal` module. The four disparity metrics — Disparate Impact Ratio, Demographic Parity Gap, Gini coefficient, and Bias Amplification Score — are computed for each `(city, period)` cell and aggregated per city, so temporal instability and cross-city divergence become visible.

Details

Reimplements the longitudinal, multi-city audit of Barman & Barman, arXiv:2603.18987. Its central lesson: bias metrics are not stable from one deployment cycle to the next and must be recomputed per period and per city.

Value

The module's audit callable returns a named `list` with the worst per-city Disparate Impact Ratio range, per-city and per-cell breakdowns, and a plain-language `interpretation`.

Examples

```
period <- c(rep("p1", 10), rep("p2", 10))
city <- rep("A", 20)
pred <- rep(c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0), 2)
grp <- rep(c(rep("X", 5), rep("Y", 5)), 2)
morie_predpol_temporal_audit(period, city, pred, grp, privileged = "X")
```

---

full_diagnostics *Full diagnostic report*

Description

Runs residual, influence, collinearity, goodness-of-fit, and specification tests, then summarises the overall assessment.

Usage

```
full_diagnostics(
  y,
  X,
  y_hat = NULL,
  model_type = "linear",
  column_names = NULL
)
```

Arguments

`y`	Response.
`X`	Design matrix.
`y_hat`	Optional fitted values (OLS used if NULL).
`model_type`	`"linear"`, `"logistic"`, `"poisson"`.
`column_names`	Optional column names for X.

Value

A `morie_diagnostic_report`.

`gam_smoother` *Thin-plate spline smoother via mgcv::gam*

Description

A penalised-spline alternative to the kernel methods above. Fits `y ~ s(x, k = k)` and returns fitted values at `x_eval`.

Usage

```
gam_smoother(x, y, x_eval = NULL, k = 10, family = stats::gaussian())
```

Arguments

`x`	Numeric covariate vector.
`y`	Numeric outcome vector.
`x_eval`	Evaluation grid (defaults to `x`).
`k`	Basis dimension for the smoother (default 10).
`family`	GLM family for `mgcv::gam` (default `gaussian()`).

Value

A list with `fit` (the fitted gam object), `x_eval`, `y_hat` (predictions), and `edf` (effective degrees of freedom).

glass_delta *Glass's delta — control-group SD denominator*

Description

Glass's delta — control-group SD denominator

Usage

```
glass_delta(x, y, control = "y", confidence = 0.95)
```

Arguments

x, y	Numeric vectors (NA dropped).
control	Which group is the control: "x" or "y" (default).
confidence	Confidence level for CI. Default 0.95.

Value

A morie_effect_size.

harmonic_mean_p *Harmonic mean p-value*

Description

For tests that may be dependent. Delegates to harmonicmeanp::p.hmp when installed; otherwise returns the raw harmonic mean (Wilson 2019 asymptotic approximation).

Usage

```
harmonic_mean_p(p_values)
```

Arguments

p_values	Numeric vector of raw p-values.

hazard_ratio_table *Hazard-ratio table from Cox model components*

Description

Hazard-ratio table from Cox model components

Usage

```
hazard_ratio_table(
  params,
  se,
  pvalues,
  confidence = 0.95,
  digits = 3L,
  apa = FALSE,
  output_format = "dataframe",
  title = "Hazard Ratios"
)
```

Arguments

params	Named numeric vector of log-HR coefficients.
se	Named numeric vector of standard errors.
pvalues	Named numeric vector of p-values.
confidence	Confidence level.
digits	Decimal places.
apa	APA formatting.
output_format	Output target.
title	Title.

hcepst *Complex cepstrum with phase unwrapping*

Description

Complex cepstrum: inverse FFT of $\log X(\omega)$ using the unwrapped phase. Unlike the real cepstrum, it preserves enough information to invert the operation, which is what enables homomorphic deconvolution.

Usage

```
hcepst(x, n_fft = NULL)
```

Arguments

`x`	Numeric vector (1-D signal).
`n_fft`	FFT length (default: next power of 2 $\geq$ `length(x)`).

Details

Reference: Oppenheim, A.V. & Schafer, R.W. (2009) *Discrete-Time Signal Processing*, 3rd ed., Pearson, chapter on cepstral analysis.

Value

List with `filtered` (complex cepstrum, real-valued), `name`, `fs`, `n_samples`, and `extra` (`quefrency`, `n_fft`, `original_length`).

Examples

```
set.seed(1)
x <- sin(2 * pi * 5 * seq(0, 1, length.out = 512))
res <- hcepst(x)
length(res$filtered)
```

`hdecon` *Homomorphic deconvolution via cepstral liftering*

Description

Separates a convolved signal $x = h * e$ into a minimum-phase impulse-response component h and an excitation e by low-time liftering of the complex cepstrum.

Usage

```
hdecon(x, cutoff, n_fft = NULL)
```

Arguments

`x`	Numeric vector (assumed convolution $h * e$).
`cutoff`	Liftering cutoff (quefrency index). Coefficients above are zeroed to isolate the slow-varying component.
`n_fft`	FFT length (default: next power of 2 $\geq$ `length(x)`).

Details

Reference: Oppenheim & Schafer (2009), *Discrete-Time Signal Processing*, 3rd ed., on homomorphic systems for convolution.

Value

List with `filtered` (minimum-phase component h), `name`, `fs`, `n_samples`, and `extra` (`excitation`, `cutoff`, `n_fft`).

Examples

```
set.seed(1)
x <- sin(2 * pi * 5 * seq(0, 1, length.out = 512))
res <- hdecon(x, cutoff = 20)
length(res$filtered)
```

`hfd` *Higuchi fractal dimension*

Description

Estimates the Higuchi (1988) fractal dimension of a 1-D time series via length scaling across `k` time-lags. Values typically fall in [1, 2]; higher values indicate greater signal complexity.

Usage

```
hfd(x, kmax = 10L)
```

Arguments

`x`	Numeric vector (length ≥ 4).
`kmax`	Maximum k (default 10).

Details

Reference: Higuchi, T. (1988) "Approach to an irregular time series on the basis of the fractal theory", *Physica D* 31(2):277–283.

Value

List with `value` (D), `name`, and `extra` (`kmax`, `n`, `L_k`).

Examples

```
set.seed(1)
x <- cumsum(rnorm(1000))
hfd(x, kmax = 10)$value
```

hierarchical_bonferroni

Hierarchical (serial gatekeeping) Bonferroni procedure

Description

Families are tested in order; if a family produces no rejections, subsequent families are blocked from testing. rmorie-specific stage-list return shape; no drop-in CRAN equivalent (**gMCP** covers the concept but with a different graphical API).

Usage

```
hierarchical_bonferroni(
  p_values_by_family,
  alpha = 0.05,
  propagate_alpha = TRUE
)
```

Arguments

`p_values_by_family`	List of numeric vectors, one per family.
`alpha`	Overall FWER level.
`propagate_alpha`	Logical; currently keeps alpha constant across families (mirrors the Python reference).

Value

A `morie_rich_result` list with one stage entry per family and an `overall_rejected` logical vector.

hochberg

Hochberg step-up FWER procedure

Description

Thin wrapper over `stats::p.adjust(method = "hochberg")`.

Usage

```
hochberg(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

`holm` *Holm step-down FWER procedure*

Description

Thin wrapper over `stats::p.adjust(method = "holm")`; uniformly more powerful than Bonferroni.

Usage

```
holm(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

`holm_sidak` *Holm-Sidak step-down procedure*

Description

Step-down Holm with Sidak's closed-form adjustment per step; equivalent to `mutoss::SidakSD` when that package is installed.

Usage

```
holm_sidak(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

hommel *Hommel FWER procedure*

Description

Thin wrapper over `stats::p.adjust(method = "hommel")`.

Usage

```
hommel(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

hosmer_lemeshow_test *Hosmer-Lemeshow goodness-of-fit test for logistic regression*

Description

Hosmer-Lemeshow goodness-of-fit test for logistic regression

Usage

```
hosmer_lemeshow_test(y, y_prob, n_groups = 10L)
```

Arguments

`y`	Binary response vector.
`y_prob`	Predicted probabilities.
`n_groups`	Number of decile groups (default 10).

Value

A `morie_specification_test`.

`hrvfd` *HRV frequency-domain metrics (VLF, LF, HF, LF/HF)*

Description

Resamples the RR-interval series uniformly at `fs_interp` Hz, estimates a Welch PSD, and integrates VLF (0.003–0.04 Hz), LF (0.04–0.15 Hz), and HF (0.15–0.40 Hz) bands.

Usage

```
hrvfd(rr, fs_interp = 4)
```

Arguments

`rr`	Numeric vector of RR intervals in milliseconds.
`fs_interp`	Uniform resampling frequency in Hz (default 4).

Details

Reference: Task Force (1996), *Circulation* 93(5):1043–1065.

Value

List with `value` (total power), `name`, and `extra` (`vlf`, `lf`, `hf`, `lf_hf_ratio`, `total_power`, `lf_norm`, `hf_norm`).

Examples

```
set.seed(1)
rr <- 800 + cumsum(rnorm(200, sd = 20))
res <- hrvfd(rr)
res$extra$lf_hf_ratio
```

`hrvnl` *HRV nonlinear metrics (Poincare SD1, SD2)*

Description

Computes the short- and long-axis standard deviations of the Poincare plot: SD1 (short-term variability) and SD2 (long-term).

Usage

```
hrvnl(rr)
```

Arguments

`rr`	Numeric vector of RR intervals in milliseconds.

Details

Reference: Brennan, M., Palaniswami, M. & Kamen, P. (2001) "Do existing measures of Poincare plot geometry reflect nonlinear features of heart rate variability?", *IEEE Trans. Biomed. Eng.* 48(11):1342–1347.

Value

List with `value` (SD1), `name`, and `extra` (`sd1`, `sd2`, `sd1_sd2_ratio`, `n_intervals`).

Examples

```
set.seed(1)
rr <- 800 + cumsum(rnorm(200, sd = 20))
res <- hrvnl(rr)
res$extra$sd1
```

`hrvtd` *HRV time-domain metrics (SDNN, RMSSD, pNN50)*

Description

Computes the standard HRV time-domain indices on an RR-interval series: SDNN, RMSSD, pNN50, mean RR, mean HR, and the HRV triangular index.

Usage

```
hrvtd(rr)
```

Arguments

`rr`	Numeric vector of RR intervals in milliseconds.

Details

Reference: Task Force (1996), *Circulation* 93(5):1043–1065.

Value

List with `value` (SDNN), `name`, and `extra` (`sdnn`, `rmssd`, `pnn50`, `mean_rr`, `mean_hr`, `hrv_triangular_index`, `n_intervals`).

Examples

```
set.seed(1)
rr <- 800 + cumsum(rnorm(200, sd = 20))
res <- hrvtd(rr)
res$extra$rmssd
```

---

i_squared *I^2 heterogeneity statistic (Higgins)*

Description

I^2 heterogeneity statistic (Higgins)

Usage

```
i_squared(estimates, standard_errors)
```

Arguments

estimates	Effect-size estimates.
standard_errors	Standard errors.

Value

Numeric I^2 percentage.

---

incidence_rate_difference *Incidence rate difference (IRD)*

Description

Incidence rate difference (IRD)

Usage

```
incidence_rate_difference(
  events1,
  person_time1,
  events2,
  person_time2,
  confidence = 0.95
)
```

Arguments

`events1`, `person_time1`
: Events and person-time in group 1.

`events2`, `person_time2`
: Events and person-time in group 2.

`confidence`
: Confidence level. Default 0.95.

Value

A `morie_effect_size`.

`intraclass_correlation` *Intraclass correlation coefficient (Shrout & Fleiss 1979)*

Description

Intraclass correlation coefficient (Shrout & Fleiss 1979)

Usage

```
intraclass_correlation(data, targets, raters, ratings, icc_type = "ICC3k")
```

Arguments

`data`	Long-format data frame.
`targets`	Subject ID column.
`raters`	Rater ID column.
`ratings`	Numeric rating column.
`icc_type`	One of ”ICC1”, ”ICC1k”, ”ICC2”, ”ICC2k”, ”ICC3”, ”ICC3k”.

`jackknife` *Delete-one (leave-one-out) jackknife*

Description

Computes the leave-one-out estimates, pseudovalues, influence values, and bias-corrected jackknife estimate. The **bootstrap** package's `bootstrap::jackknife` is the canonical CRAN reference; it is invoked when installed and the rmorie-shape result is reconstructed around it. Falls back to an inline loop otherwise.

Usage

```
jackknife(data, statistic, ci_level = 0.95)
```

Arguments

`data`	Numeric vector or matrix.
`statistic`	Function returning a scalar.
`ci_level`	Confidence level.

Value

A `morie_jackknife_result`.

See Also

`bootstrap::jackknife`, `resample::jackknife`.

`kde` *Kernel density estimation*

Description

Computes f-hat(x) equal to one over n times h times the sum over i of K of (x minus X_i) divided by h.

Usage

```
kde(x, x_eval, bandwidth, kernel_type = KERNEL_GAUSSIAN)
```

Arguments

`x`	Numeric data vector.
`x_eval`	Evaluation grid.
`bandwidth`	Positive bandwidth.
`kernel_type`	Integer code or kernel name.

Value

Numeric vector of estimated densities.

References

Silverman, B. W. (1986). Density Estimation for Statistics and Data Analysis. Chapman and Hall.

kendall_correlation *Kendall's tau-b correlation*

Description

Kendall's tau-b correlation

Usage

```
kendall_correlation(x, y)
```

Arguments

x, y	Numeric vectors.

kernel-codes *Kernel type integer codes*

Description

Integer codes used by morie's C++ semiparametric bridge to select the kernel function for local-polynomial smoothing. Mirror the Python `morie.semipar.KernelType` enum so an R caller can pass these constants directly to any C++ kernel routine.

Usage

```
KERNEL_GAUSSIAN

KERNEL_EPANECHNIKOV

KERNEL_UNIFORM

KERNEL_TRIANGULAR

KERNEL_BIWEIGHT
```

Format

Integer scalars (0L, 1L, 2L, 3L, 4L).

An object of class `integer` of length 1.

An object of class `integer` of length 1.

An object of class `integer` of length 1.

An object of class `integer` of length 1.

An object of class `integer` of length 1.

Details

- `KERNEL_GAUSSIAN`: $K(u) = (1/\sqrt{2\pi})\exp(-u^2/2)$
- `KERNEL_EPANECHNIKOV`: $K(u) = (3/4)(1-u^2)$ on |u| <= 1
- `KERNEL_UNIFORM`: $K(u) = 1/2$ on |u| <= 1
- `KERNEL_TRIANGULAR`: $K(u) = 1 - |u|$ on |u| <= 1
- `KERNEL_BIWEIGHT`: $K(u) = (15/16)(1-u^2)^2$ on |u| <= 1

`kernel_cond_moments` *Kernel-weighted conditional mean and variance*

Description

Useful for the conditional outcome stage of TMLE / AIPW.

Usage

```
kernel_cond_moments(x, y, x_eval, bandwidth, return_variance = TRUE)
```

Arguments

`x`	Numeric covariate vector.
`y`	Numeric outcome vector.
`x_eval`	Evaluation grid.
`bandwidth`	Positive bandwidth.
`return_variance`	Logical; if FALSE, only the mean is returned.

Value

Either a numeric vector (mean only) or a list with `mean` and `variance`.

`kernel_eval` *Evaluate a kernel function at point u*

Description

Evaluate a kernel function at point u

Usage

```
kernel_eval(u, kernel_type = KERNEL_GAUSSIAN)
```

Arguments

`u`	Numeric evaluation point (scaled by bandwidth).
`kernel_type`	Integer code or kernel name. One of `KERNEL_GAUSSIAN` (0), `KERNEL_EPANECHNIKOV` (1), `KERNEL_UNIFORM` (2), `KERNEL_TRIANGULAR` (3), `KERNEL_BIWEIGHT` (4), or the matching string.

Value

Kernel density value K(u).

`kfd` *Katz fractal dimension*

Description

Katz fractal dimension $D = log_{10}(n-1)/(log_{10}(n-1) + log_{10}(d/L))$ of a 1-D signal. L is total path length and d is the diameter (max distance from the first sample).

Usage

```
kfd(x)
```

Arguments

`x`	Numeric vector.

Details

Reference: Katz, M.J. (1988) "Fractals and the analysis of waveforms", *Comput. Biol. Med.* 18(3):145–156.

Value

List with `value` (D), `name`, and `extra` (`L`, `d`, `n`).

Examples

```
set.seed(1)
x <- cumsum(rnorm(1000))
res <- kfd(x)
res$value
```

`ks_test_one_sample` *One-sample Kolmogorov-Smirnov test*

Description

One-sample Kolmogorov-Smirnov test

Usage

```
ks_test_one_sample(x, cdf = "pnorm", args = list())
```

Arguments

`x`	Numeric vector.
`cdf`	Name of a CDF function (e.g. "pnorm", "pexp"). Defaults to "pnorm". A bare distribution name like "norm" is auto-prefixed.
`args`	List of extra arguments to pass to `cdf`.

`ks_test_two_sample` *Two-sample Kolmogorov-Smirnov test*

Description

Two-sample Kolmogorov-Smirnov test

Usage

```
ks_test_two_sample(x, y)
```

Arguments

`x`, `y`	Numeric vectors.

leave_one_out_cv *Leave-one-out cross-validation*

Description

Convenience wrapper that calls `.boot_cross_validate()` with `n_folds = length(y)`. `rsample::loo_cv` is the tidymodels equivalent (cross-referenced).

Usage

```
leave_one_out_cv(X, y, model_fn, score_fn)
```

Arguments

`X`	Numeric matrix or data.frame of predictors.
`y`	Numeric or factor outcome vector aligned with rows of `X`.
`model_fn`	Function `(X, y) -> fitted-model` used on each training fold.
`score_fn`	Function `(y_true, y_pred) -> numeric` returning a single performance metric.

Value

A `morie_cv_result`.

See Also

`rsample::loo_cv`, `caret::trainControl`.

levene_test *Levene's test for equality of variances*

Description

Levene's test for equality of variances

Usage

```
levene_test(..., center = "median")
```

Arguments

`...`	Two or more numeric vectors.
`center`	One of "median" (Brown-Forsythe), "mean", "trimmed".

license_check *Runtime license-compatibility guard for morie*

Description

R parity of `morie._license_check`. Exposes the FSF GPL-compatible licence list and a `check_plugin_license()` helper that downstream R packages / plugins can call to confirm GPL compatibility before linking against morie internals. The guard is advisory — it warns or raises but does not enforce at the R-namespace level. For stronger guarantees see the companion userspace LSM-style daemon (`daemon/morie_lsm.py`) and the kernel companion module (`kernel-module/morie.c`).

Value

`morie_gpl_compatible_licenses()` returns a character vector of GPL-compatible SPDX identifiers; `check_plugin_license()` returns a logical (invisibly), signalling a warning or error when the supplied licence is not GPL-compatible.

Examples

```
morie_gpl_compatible_licenses()
```

likelihood_ratio_test *Likelihood ratio test for nested models*

Description

Likelihood ratio test for nested models

Usage

```
likelihood_ratio_test(ll_restricted, ll_full, df_diff)
```

Arguments

`ll_restricted`	Log-likelihood of the restricted model.
`ll_full`	Log-likelihood of the full model.
`df_diff`	Difference in degrees of freedom.

Value

A `morie_specification_test`.

lilliefors_test *Lilliefors test for normality*

Description

Uses **nortest::lillie.test** when available; otherwise falls back to a plain KS test with estimated parameters (p-value approximate).

Usage

```
lilliefors_test(x)
```

Arguments

x	Numeric vector.

link_test *Pregibon link test*

Description

Pregibon link test

Usage

```
link_test(y, X, model_type = "linear")
```

Arguments

y	Response.
X	Design matrix.
model_type	"linear" or "logistic".

Value

A morie_specification_test.

`local_fdr` *Local false discovery rate via empirical-Bayes two-component mixture*

Description

Estimates the local FDR for each test as $lfdr_i = \pi_0 \, f_0(z_i) / f(z_i)$, where $z_i = \Phi^{-1}(1 - p_i/2)$ are two-sided z-scores, f_0 is the standard-normal null density, f is a kernel density estimate of the observed z-scores, and π_0 is the proportion of null hypotheses estimated by the Storey-style cutoff at $p > 0.5$.

Usage

```
local_fdr(p_values, pi0_method = "bootstrap", labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values in ‘$[0, 1]$‘.
`pi0_method`	Pi-zero estimator. Accepted: `"bootstrap"` (alias for the Storey-style cutoff at 0.5; retained for API parity with the Python sibling).
`labels`	Optional character vector of test labels.

Value

A data frame with columns `p_value`, `z_score`, `local_fdr`, and (if supplied) `label`. The data frame additionally carries class `morie_rich_result`.

Examples

```
set.seed(1)
p <- c(stats::runif(80),
       stats::pnorm(-abs(stats::rnorm(20, mean = 3))) * 2)
lfdr <- local_fdr(p)
head(lfdr)
```

`local_linear` *Local linear kernel regression*

Description

Avoids the boundary bias of Nadaraya-Watson by fitting a local linear model at each evaluation point.

Usage

```
local_linear(x, y, x_eval, bandwidth, return_slope = FALSE)
```

Arguments

`x`	Numeric covariate vector.
`y`	Numeric outcome vector.
`x_eval`	Evaluation grid.
`bandwidth`	Positive bandwidth.
`return_slope`	Logical; if TRUE, also return local slopes.

Value

If `return_slope = FALSE`, a numeric vector of fitted values; otherwise a list with `y_hat` and `beta_hat`.

References

Fan, J. and Gijbels, I. (1996). Local Polynomial Modelling and Its Applications. Chapman and Hall.

`longitudinal_sim` *Synchronised longitudinal-panel simulation (R parity)*

Description

Clean-room R parity of `morie.longitudinal_sim` for synchronised multivariate longitudinal-panel simulation. Implements SyncRNG, VAR coefficient generation with stationarity preservation, MVN draws under structured covariance kernels, and tidy panel output.

Details

Clean-room note: this module re-implements the techniques used in the Hlozek–Bangari Collaborative-CIFAR-Catalyst project (https://github.com/bangari-19/Collaborative-CIFAR-Project-) without copying any source. That repository is unlicensed. The techniques themselves — synchronised PRNG streams, lagged AR coefficient matrices, multivariate normal generation under Toeplitz / compound-symmetric covariance — are standard methods from Hamilton (1994) and Diggle, Liang, Zeger (1994), implemented here independently.

Value

The simulation callables return tidy longitudinal-panel `data.frame`s; `morie_sync_rng()` returns an environment exposing synchronised `rnorm`, `runif`, and `sample` methods.

Examples

```
rng <- morie_sync_rng(42)
```

loocv_bandwidth *Leave-one-out cross-validation bandwidth for NW regression*

Description

Minimises CV(h) equal to one over n times sum over i of (Y_i minus m-hat-h-minus-i of X_i) squared on a grid spanning `bw_min` to `bw_max`.

Usage

```
loocv_bandwidth(x, y, bw_min = NULL, bw_max = NULL, n_grid = 30L)
```

Arguments

`x`	Numeric covariate vector.
`y`	Numeric outcome vector.
`bw_min`	Minimum candidate bandwidth (defaults to 0.1 times the Silverman bandwidth).
`bw_max`	Maximum candidate bandwidth (defaults to 2.0 times the Silverman bandwidth).
`n_grid`	Number of candidate values.

Value

Optimal bandwidth (numeric scalar).

References

Hardle, W. (1990). Applied Nonparametric Regression. Cambridge.

mann_whitney_u *Mann-Whitney U / Wilcoxon rank-sum test*

Description

Mann-Whitney U / Wilcoxon rank-sum test

Usage

```
mann_whitney_u(x, y, alternative = "two.sided")
```

Arguments

`x, y`	Numeric vectors.
`alternative`	One of "two.sided", "less", "greater".

manski_bounds *Manski worst-case bounds for the ATE*

Description

Under no assumptions about selection, the ATE is only partially identified. Returns a named list with `lower_bound`, `upper_bound`, `point_estimate`, `width`.

Usage

```
manski_bounds(
  outcome_treated,
  outcome_control,
  p_treated,
  outcome_range = NULL
)
```

Arguments

`outcome_treated`	Outcomes for treated units.
`outcome_control`	Outcomes for control units.
`p_treated`	Proportion treated.
`outcome_range`	c(min, max) on the outcome. Default c(0, 1).

Value

Named list.

mcnemar_test *McNemar's test (paired nominal data)*

Description

McNemar's test (paired nominal data)

Usage

```
mcnemar_test(contingency_table, exact = FALSE)
```

Arguments

`contingency_table`	2x2 table.
`exact`	Use exact binomial.

`midranks` *Midrank vector with tie summary (Gibbons Ch 5.6.2)*

Description

Identical to `rank(x, ties.method = "average")` plus a tie- correction term `sum t_j^3 - t_j` over tied groups.

Usage

```
midranks(x)
```

Arguments

`x`	Numeric vector.

Value

Named list: midranks, n, ties, tie_correction.

Examples

```
midranks(x = rnorm(50))
```

`model_comparison_table` *Compare multiple model fits on AIC, BIC, log-likelihood and (optional) LR tests*

Description

Compare multiple model fits on AIC, BIC, log-likelihood and (optional) LR tests

Usage

```
model_comparison_table(
  models,
  nested = FALSE,
  digits = 3L,
  output_format = "dataframe",
  title = "Model Comparison"
)
```

Arguments

`models`	Named list of fitted models.
`nested`	If TRUE, run LR tests against the previous model in the list.
`digits`	Decimal places.
`output_format`	Output target.
`title`	Title.

`morie_anominate_ideal_points` *Alpha-NOMINATE ideal-point estimation via* **anominate**

Description

Thin extender over `anominate::anominate` for Bayesian alpha-NOMINATE ideal-point estimation on roll-call legislative data (Carroll, Lewis, Lo, Poole & Rosenthal, 2013).

Usage

```
morie_anominate_ideal_points(rcObject, ...)
```

Arguments

`rcObject`	A roll-call object as built by `pscl::rollcall`, suitable for passing to `anominate::anominate`.
`...`	Further arguments forwarded to `anominate::anominate` (e.g. `dims`, `nsamp`, `thin`, `burnin`, `minvotes`, `lop`, `polarity`, `random.starts`, `verbose`).

Value

A list with `$method = "anominate::anominate"` and `$raw` (an `anominate` posterior-sample object).

Examples

```
## Not run:
  if (requireNamespace("anominate", quietly = TRUE) &&
      requireNamespace("pscl", quietly = TRUE)) {
    data("sen90", package = "anominate")
    morie_anominate_ideal_points(sen90, dims = 1, nsamp = 200, burnin = 100)
  }

## End(Not run)
```

morie_anova_one_way *One-way ANOVA*

Description

One-way ANOVA

Usage

```
morie_anova_one_way(...)
```

Arguments

...	Numeric vectors, one per group.

Value

Named list: `F`, `df_between`, `df_within`, `p_value`, `eta_squared`.

Examples

```
morie_anova_one_way(rnorm(30, 0), rnorm(30, 0.5), rnorm(30, 1))
```

morie_arch_in_mean *ARCH(1)-in-mean model*

Description

ARCH(1)-in-mean model

Usage

```
morie_arch_in_mean(x)
```

Arguments

x	Numeric return series.

Value

Named list with `mu`, `delta`, `omega`, `alpha`, `loglik`, `conditional_variance`, `n`, `method`.

Examples

```
morie_arch_in_mean(x = rnorm(50))
```

`morie_arsau_analyze_aggregate_summary` *Analysis of the ARSAU aggregate-summary-by-year file (2020-2022).*

Description

The aggregate file is a long-format `YEAR_2020 / YEAR_2021 / YEAR_2022` panel keyed by `(SECTION, CATEGORY, UNITS OF MEASURE)`. This function rebuilds the implied time series, runs year-on-year change against the `REPORT_SCOPE` rows (the headline volume series), and surfaces a data-quality audit.

Builds an implied YoY series from the YEAR_2020 / YEAR_2021 / YEAR_2022 columns against the REPORT_SCOPE headline volume row.

Usage

```
morie_arsau_analyze_aggregate_summary(
  year_range = "2020-2022",
  language = "en",
  data_dir = NULL
)

morie_arsau_analyze_aggregate_summary(
  year_range = "2020-2022",
  language = "en",
  data_dir = NULL
)
```

Arguments

`year_range`	"2020-2022".
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

Value

A list classed `c("morie_arsau_result", "morie_rich_result", "list")`.

References

Ontario Ministry of the Solicitor General, ARSAU 2020-2022 aggregate-summary-by-year technical notes.

`morie_arsau_analyze_detailed_dataset` *Wide-format analysis of the 2020-2022 detailed-incident dataset.*

Description

Chains:

- `mrm_uof_force_concentration` on `POLICE_SERVICE`
- `mrm_uof_weapon_diversity` on `POLICE_SERVICE x ASSIGNMENT_TYPE`
- `mrm_uof_yoy_change` on `REPORTING_YEAR`
- `mrm_uof_data_quality_audit`

Usage

```
morie_arsau_analyze_detailed_dataset(
  year_range = "2020-2022",
  language = "en",
  data_dir = NULL
)

morie_arsau_analyze_detailed_dataset(
  year_range = "2020-2022",
  language = "en",
  data_dir = NULL
)
```

Arguments

`year_range`	"2020-2022".
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

Value

A list classed `c("morie_arsau_result", "morie_rich_result", "list")`.

References

Ontario Ministry of the Solicitor General, ARSAU 2020-2022 detailed_dataset technical notes.

`morie_arsau_analyze_individual_records` *End-to-end analysis of the ARSAU individual_records CSV for one year.*

Description

Chains demographic-disparity tests over Race, Gender, and AgeCategory against the `IndivInjuries_PhysicalInj` outcome column, plus a data-quality audit against the sidecar.

Chains demographic disparity by Race, Gender, AgeCategory against the IndivInjuries_PhysicalInjuries outcome (Yes/No coerced). Tolerates the 2023 trailing-space typo in the outcome column.

Usage

```
morie_arsau_analyze_individual_records(
  year,
  language = "en",
  data_dir = NULL,
  bootstrap_reps = 0L
)

morie_arsau_analyze_individual_records(
  year,
  language = "en",
  data_dir = NULL,
  bootstrap_reps = 0L
)
```

Arguments

`year`	2023 or 2024.
`language`	”en” or ”fr”.
`data_dir`	Optional explicit ARSAU root.
`bootstrap_reps`	Forwarded to `mrm_uof_demographic_disparity`.

Value

A list classed `c("morie_arsau_result", "morie_rich_result", "list")`.

References

Ontario Ministry of the Solicitor General, ARSAU 2023 and 2024 individual_records technical release notes.

`morie_arsau_analyze_main_records` *End-to-end analysis of the ARSAU main_records CSV for one year.*

Description

Chains:

- `mrm_uof_force_concentration` over `PoliceService`
- `mrm_uof_weapon_diversity` over `IncidentType x PoliceService`
- `mrm_uof_data_quality_audit` against the published CKAN sidecar (when present)

Chains `mrm_uof_force_concentration` (PoliceService), `mrm_uof_weapon_diversity` (IncidentType x PoliceService), and `mrm_uof_data_quality_audit` (against the CKAN sidecar).

Usage

```
morie_arsau_analyze_main_records(year, language = "en", data_dir = NULL)

morie_arsau_analyze_main_records(year, language = "en", data_dir = NULL)
```

Arguments

`year`	2023 or 2024.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

Details

Region-locality is NOT meaningful for main_records – only the `OPP_PoliceService_Region` column is published, and it pairs one column with itself. See `morie_arsau_analyze_detailed_dataset` for the 2020-2022 layout that exposes more region columns.

Value

A list classed `c("morie_arsau_result", "morie_rich_result", "list")`.

References

Ontario Ministry of the Solicitor General, ARSAU 2023 and 2024 main_records technical release notes.

`morie_arsau_analyze_probe_cycle_records`
Analysis of ARSAU probe_cycle_records (CEW telemetry).

Description

The probe-cycle file is intentionally narrow (BatchFileName + Indiv_Index + a comma-separated cycle string). This function computes the cycle-count distribution per incident and runs a data-quality audit.

Usage

```
morie_arsau_analyze_probe_cycle_records(year, language = "en", data_dir = NULL)

morie_arsau_analyze_probe_cycle_records(year, language = "en", data_dir = NULL)
```

Arguments

`year`	2023 or 2024.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

Value

A list classed `c("morie_arsau_result", "morie_rich_result", "list")`.

References

Ontario Ministry of the Solicitor General, ARSAU probe_cycle_records technical notes (2023 and 2024).

`morie_arsau_analyze_weapon_records`
Analysis of ARSAU weapon_records.

Description

Chains `mrm_uof_weapon_diversity` over `Weapon x Location` (the only two categorical columns the file publishes) plus a Weapon-only frequency table plus a data-quality audit.

Chains Weapon x Location chi-square + weapon frequency table + DQ audit. 2023 needs `allow_invalid = TRUE`.

Usage

```
morie_arsau_analyze_weapon_records(
  year,
  allow_invalid = FALSE,
  language = "en",
  data_dir = NULL
)

morie_arsau_analyze_weapon_records(
  year,
  allow_invalid = FALSE,
  language = "en",
  data_dir = NULL
)
```

Arguments

`year`	2023 or 2024.
`allow_invalid`	See `morie_arsau_load_weapon_records`.
`language`	”en” or ”fr”.
`data_dir`	Optional explicit ARSAU root.

Details

The 2023 file is the ministry-flagged-invalid release and requires `allow_invalid = TRUE`.

Value

A list classed `c("morie_arsau_result", "morie_rich_result", "list")`.

References

Ontario Ministry of the Solicitor General, ARSAU 2023 and 2024 weapon_records technical notes – the 2023 release accompanies an explicit invalidity flag.

`morie_arsau_available_datasets` *List ARSAU dataset kinds, optionally restricted to one year.*

Description

List ARSAU dataset kinds, optionally restricted to one year.

Usage

```
morie_arsau_available_datasets(year = NULL, language = "en", data_dir = NULL)
```

Arguments

year	Optional year; NULL lists everything.
language	"en" or "fr".
data_dir	Optional explicit ARSAU root.

`morie_arsau_available_years` *List ARSAU year / year-range buckets.*

Description

List ARSAU year / year-range buckets.

Usage

```
morie_arsau_available_years(data_dir = NULL, language = "en")
```

Arguments

data_dir	Optional explicit ARSAU root.
language	"en" or "fr".

`morie_arsau_ckan_url` *Build the upstream CKAN* `datastore_search` *URL for a registry entry.*

Description

Returns `NA_character_` for entries that do not publish a sidecar (e.g. the 2023 weapon_records release).

Usage

```
morie_arsau_ckan_url(kind, year, limit = 5000L)
```

Arguments

kind	One of `ARSAU_KINDS()`.
year	One of `ARSAU_YEARS()`.
limit	Integer; CKAN `limit` parameter. Default 5000.

Value

Character scalar URL, or `NA_character_`.

References

Ontario Data Catalogue CKAN API: `datastore_search` (https://data.ontario.ca/).

`morie_arsau_describe` *Describe a single ARSAU dataset entry.*

Description

Describe a single ARSAU dataset entry.

Usage

```
morie_arsau_describe(
  kind,
  year,
  language = "en",
  data_dir = NULL,
  n_preview_rows = 3L
)
```

Arguments

`kind`	One of `ARSAU_KINDS()`.
`year`	One of `ARSAU_YEARS()`.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.
`n_preview_rows`	Number of rows from the CSV head to include.

`morie_arsau_download` *Bulk-download every ARSAU CSV + sidecar from the upstream Catalogue.*

Description

This is the R-side equivalent of running the maintainer's `scripts/refresh_arsau.py` mirror — a non-trivial pipeline that walks the CKAN package, follows per-resource redirects, handles rate-limits, verifies SHA digests against the published values, and lands the files under `MORIE_ARSAU_DIR`. Porting it requires an end-to-end retry + checksum manager that does not yet have a tested R analogue; per the morie maintenance policy, network bulk fetches must be reproducible across CRAN test environments before the wrapper is exposed. Stubbed for now.

Usage

```
morie_arsau_download(target_dir, ...)
```

Arguments

`target_dir`	Destination directory.
`...`	Reserved.

Value

Stops with `NotYetPorted`.

`morie_arsau_fetch_sidecar` *Fetch the CKAN sidecar JSON for a registry entry.*

Description

Optional helper. Requires **httr2** (and **jsonlite** via the existing `morie_arsau_read_sidecar` contract).

Usage

```
morie_arsau_fetch_sidecar(kind, year, limit = 5000L, timeout_sec = 30L)
```

Arguments

`kind`	One of `ARSAU_KINDS()`.
`year`	One of `ARSAU_YEARS()`.
`limit`	Integer; CKAN `limit` parameter. Default 5000.
`timeout_sec`	Request timeout in seconds. Default 30.

Value

A list with `fields` and `records`, ready for `morie_arsau_sidecar_schema` / `morie_arsau_sidecar_to_frame`.

References

Ontario Data Catalogue CKAN API.

`morie_arsau_load_aggregate_summary` *Load ARSAU aggregate-summary-by-year CSV (2020-2022 only).*

Description

Load ARSAU aggregate-summary-by-year CSV (2020-2022 only).

Usage

```
morie_arsau_load_aggregate_summary(
  year_range = "2020-2022",
  language = "en",
  data_dir = NULL
)
```

Arguments

`year_range`	"2020-2022".
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

`morie_arsau_load_detailed_dataset` *Load ARSAU detailed-incident-level CSV (2020-2022 only).*

Description

Load ARSAU detailed-incident-level CSV (2020-2022 only).

Usage

```
morie_arsau_load_detailed_dataset(
  year_range = "2020-2022",
  language = "en",
  data_dir = NULL
)
```

Arguments

`year_range`	"2020-2022".
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

`morie_arsau_load_individual_records`
Load ARSAU individual_records CSV.

Description

Load ARSAU individual_records CSV.

Usage

```
morie_arsau_load_individual_records(year, language = "en", data_dir = NULL)
```

Arguments

`year`	2023 or 2024.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

`morie_arsau_load_main_records`
Load ARSAU main_records CSV for the given year.

Description

Load ARSAU main_records CSV for the given year.

Usage

```
morie_arsau_load_main_records(year, language = "en", data_dir = NULL)
```

Arguments

`year`	2023 or 2024.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

morie_arsau_load_probe_cycle_records

Load ARSAU probe_cycle_records CSV (CEW telemetry).

Description

Load ARSAU probe_cycle_records CSV (CEW telemetry).

Usage

```
morie_arsau_load_probe_cycle_records(year, language = "en", data_dir = NULL)
```

Arguments

`year`	2023 or 2024.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

morie_arsau_load_weapon_records

Load ARSAU weapon_records CSV.

Description

2023 requires `allow_invalid = TRUE` (ministry-flagged invalid).

Usage

```
morie_arsau_load_weapon_records(
  year,
  allow_invalid = FALSE,
  language = "en",
  data_dir = NULL
)
```

Arguments

`year`	2023 or 2024.
`allow_invalid`	Logical; required `TRUE` for 2023.
`language`	"en" or "fr".
`data_dir`	Optional explicit ARSAU root.

`morie_arsau_read_markdown_dictionary`
Read an Ontario-Catalogue Markdown data-dictionary sidecar.

Description

Parses a simple pipe-table Markdown sidecar of the form

```
| name | type | notes |
|------|------|-------|
| foo  | int  | ...   |
```

as published by some ARSAU releases. No external dependencies are required: the parser is pure base R.

Usage

```
morie_arsau_read_markdown_dictionary(path)
```

Arguments

`path`	Path to the Markdown file.

Value

A `data.frame` with one row per table row. Returns an empty `data.frame` if the file has no parseable pipe table.

References

Ontario Ministry of the Solicitor General data dictionaries accompanying the ARSAU CSV releases.

`morie_arsau_read_sidecar`
Read a CKAN datastore_search JSON sidecar.

Description

Handles both bare `{fields, records}` and the `{result: {fields, records}}` wrapper shape.

Usage

```
morie_arsau_read_sidecar(path)
```

Arguments

`path`	Path to the JSON file.

Value

Named list with `fields` and `records`.

`morie_arsau_read_xlsx_dictionary` *Read an Ontario-Catalogue XLSX data-dictionary sidecar.*

Description

Some ARSAU releases ship a companion `*.xlsx` file alongside the CSV that holds the column-level data dictionary (variable name

- dtype + notes). This helper reads the first sheet via **readxl** and normalises the column names to `name / type / notes`. Requires the optional **readxl** dependency.

Usage

```
morie_arsau_read_xlsx_dictionary(path, sheet = 1L)
```

Arguments

`path`	Path to the XLSX file.
`sheet`	Sheet identifier (name or 1-based integer). Default `1L`.

Value

A `data.frame` with columns `name`, `type`, `notes`. Other columns from the XLSX are preserved with their upstream names.

References

Ontario Ministry of the Solicitor General data dictionaries accompanying the ARSAU CSV releases.

`morie_arsau_registry_df`
ARSAU registry rendered as a tidy `data.frame`.

Description

Returns one row per `(year_or_range, kind)` entry in the package's internal registry, with the same columns as the Python `ARSAU_REGISTRY` mapping but in row-major `data.frame` form. The underlying list-of-lists is still available via `ARSAU_REGISTRY`.

Usage

```
morie_arsau_registry_df(language = "en")
```

Arguments

`language`	"en" or "fr"; selects the description column.

Value

A `data.frame` with columns `year_or_range`, `kind`, `csv_filename`, `sidecar_filename`, `expected_rows`, `expected_cols`, `is_valid`, `description`.

References

Ontario Ministry of the Solicitor General, ARSAU per-resource technical release notes (2020-2022 / 2023 / 2024).

`morie_arsau_sidecar_schema`
Extract a simplified `[name, type, notes]` *schema from a parsed CKAN sidecar.*

Description

Accepts the result of `morie_arsau_read_sidecar` (a list with `fields` and `records` entries) and returns a tidy `data.frame` of column metadata. Entries that lack an `id` are dropped.

Usage

```
morie_arsau_sidecar_schema(sidecar)
```

Arguments

`sidecar`	A list as returned by `morie_arsau_read_sidecar()`.

Value

A `data.frame` with columns `name`, `type`, `notes`.

References

CKAN *datastore_search* response schema, as served by `datastore_search` (https://data.ontario.ca/).

`morie_arsau_sidecar_to_frame` *Convert a CKAN sidecar's* `records` *array-of-arrays into a* `data.frame`.

Description

The `fields[].id` array supplies the column names; records are array-of-array, so the column order in the JSON matches the column order in the resulting `data.frame`.

Usage

```
morie_arsau_sidecar_to_frame(sidecar)
```

Arguments

`sidecar`	A list as returned by `morie_arsau_read_sidecar()`.

Value

A `data.frame` (zero rows if `records` is empty).

References

CKAN *datastore_search* response schema.

`morie_audit_all_variables` *Audit both OTIS and ARSAU.*

Description

Audit both OTIS and ARSAU.

Usage

```
morie_audit_all_variables(otis_specs = NULL, arsau_specs = NULL)
```

Arguments

`otis_specs`, `arsau_specs`
: See per-domain functions.

Value

Named list with `$otis` and `$arsau` audit results.

`morie_audit_arsau_variables`
: *Audit every ARSAU variable.*

Description

Audit every ARSAU variable.

Usage

```
morie_audit_arsau_variables(dataset_specs = NULL)
```

Arguments

`dataset_specs` See `morie_audit_otis_variables`.

Value

A list with class `morie_audit_result`.

`morie_audit_otis_variables`
: *Audit every OTIS variable.*

Description

For each OTIS dataset, this function expects a list of column specifications. By default it constructs the specs from the columns in `DATASET_REGISTRY` (in R, these are stored on the Python side via the dictionary parser; on the R side we fall back to a minimal name-only list and rely on the heuristic classifier when dtype/valid_values are unknown).

Usage

```
morie_audit_otis_variables(dataset_specs = NULL)
```

Arguments

`dataset_specs` Optional list keyed by dataset id, each entry a list of `list(name, dtype, valid_values)` entries. When `NULL`, the function uses a built-in minimal spec extracted from the existing R-side OTIS metadata.

Details

For a richer audit that consults the bilingual XLSX dictionary, use the Python module `morie.audit_variables`.

Value

A list with class `morie_audit_result`.

`morie_audit_public_outputs` *Audit declared outputs against files on disk*

Description

Audit declared outputs against files on disk

Usage

```
morie_audit_public_outputs(project_root = NULL, manifest = NULL)
```

Arguments

`project_root`	Project root directory.
`manifest`	Manifest data frame. If `NULL`, loaded from disk.

Value

Data frame containing declared and observed output status.

Examples

```
# Craft a tempdir manifest + output file, then audit:
tdir <- tempfile("morie-doc-")
dir.create(tdir)
writeLines("x,y\n1,2", file.path(tdir, "results.csv"))
man <- data.frame(
  output = "results.csv",
  public_path = file.path(tdir, "results.csv"),
  size_kb = 0.01, modified = format(Sys.Date())
)
morie_audit_public_outputs(project_root = tdir, manifest = man)
```

`morie_bayes_cpi_genomic` *BayesC-pi spike-and-slab variable selection (short Gibbs)*

Description

BayesC-pi spike-and-slab variable selection (short Gibbs)

Usage

```
morie_bayes_cpi_genomic(
  x,
  y,
  n_iter = 300,
  burn = 100,
  pi_init = 0.1,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	(n x p) marker matrix.
`y`	Numeric response.
`n_iter`	Iterations.
`burn`	Burn-in.
`pi_init`	Initial inclusion probability.
`seed`	Seed.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("bglup", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, beta, beta_pip, pi, sigma_b2, sigma2, n_iter, n, p, method).

References

Habier-Fernando-Kizilkaya-Garrick (2011); Montesinos Lopez Ch 4.

Examples

```
morie_bayes_cpi_genomic(x = rnorm(50), y = rnorm(50))
```

`morie_bayes_ridge_gibbs` *BayesA via short Gibbs sampler (Meuwissen-Hayes-Goddard 2001)*

Description

Per-marker variance with scaled inverse chi-squared prior.

Usage

```
morie_bayes_ridge_gibbs(
  x,
  y,
  n_iter = 200,
  burn = 50,
  df0 = 4,
  S0 = NULL,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	(n x p) marker matrix.
`y`	Numeric response.
`n_iter`	Iterations.
`burn`	Burn-in.
`df0`	Prior df (default 4).
`S0`	Prior scale (default anchors to var(y)/p).
`seed`	Seed.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("brdgf", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, beta, beta_se, sigma_j2, sigma2, n_iter, n, p, method).

References

Meuwissen-Hayes-Goddard (2001) Genetics 157:1819.

Examples

```
morie_bayes_ridge_gibbs(x = rnorm(50), y = rnorm(50))
```

`morie_bayesian_lasso_full` *Bayesian LASSO (Park & Casella 2008 short Gibbs)*

Description

Bayesian LASSO (Park & Casella 2008 short Gibbs)

Usage

```
morie_bayesian_lasso_full(
  x,
  y,
  n_iter = 200,
  burn = 50,
  lam = NULL,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	(n x p) marker matrix.
`y`	Numeric response.
`n_iter`	Total iterations (default 200).
`burn`	Burn-in (default 50).
`lam`	Optional fixed lambda (else empirical-Bayes updated).
`seed`	Random seed.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("blasf", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, beta, intercept, se, beta_se, lam, sigma2, n_iter, n, p, method).

References

Park & Casella (2008) JASA 103:681. Montesinos Lopez Ch 4.

Examples

```
morie_bayesian_lasso_full(
  x = matrix(rnorm(150), 50, 3), y = rnorm(50),
  n_iter = 50L, burn = 10L, lam = 1, seed = 1L,
  deterministic_seed = TRUE
)
```

morie_bayesian_ridge_regression

Bayesian ridge regression (RR-BLUP closed form)

Description

beta_hat = solve(X'X + lambda*I) %% X'y

Usage

```
morie_bayesian_ridge_regression(x, y, lam = NULL)
```

Arguments

x	(n x p) marker matrix.
y	Numeric response.
lam	Optional ridge parameter; default Endelman rrBLUP-style.

Value

list(estimate, beta, intercept, se, beta_se, lam, n, p, method).

References

Montesinos Lopez Ch 4.

Examples

```
morie_bayesian_ridge_regression(x = rnorm(50), y = rnorm(50))
```

morie_boot_basic_ci

Direct bridge to `boot::boot.ci`

Description

Thin pass-through that computes bootstrap confidence intervals from a `boot` object via `boot::boot.ci`, returning a tidy named list with `(ci_lower, ci_upper)` per requested type.

Usage

```
morie_boot_basic_ci(
  boot_obj,
  type = c("perc", "bca", "basic", "norm", "stud"),
  conf = 0.95
)
```

Arguments

`boot_obj`	A `boot` object as returned by `morie_boot_run()` or `boot::boot`.
`type`	Character vector of CI types; any of `"perc"`, `"bca"`, `"basic"`, `"norm"`, `"stud"`.
`conf`	Confidence level (default 0.95).

Value

Named list of length `length(type)`; each element is a numeric length-2 vector `c(ci_lower, ci_upper)`.

See Also

`boot::boot.ci`.

`morie_boot_run` *Direct bridge to* `boot::boot`

Description

Thin pass-through that adapts an rmorie-style `statistic(data) -> scalar` callable to `boot::boot`'s `statistic(data, indices) -> scalar` signature and returns the raw `boot` object. Useful when the caller wants to use `boot`'s downstream helpers (`boot::boot.ci`, `boot::tilt.boot`, `boot::jack.after.boot`) directly.

Usage

```
morie_boot_run(data, statistic, R = 2000L, strata = NULL, ...)
```

Arguments

`data`	A numeric vector, matrix, or data.frame.
`statistic`	Function `f(data) -> scalar`.
`R`	Number of bootstrap replicates.
`strata`	Optional integer stratification vector.
`...`	Forwarded to `boot::boot`.

Value

A `boot` object as returned by `boot::boot`.

See Also

`boot::boot`, `morie_boot_basic_ci()`.

`morie_bootstrap_sample`
Bootstrap resampling for any statistic

Description

Bootstrap resampling for any statistic

Usage

```
morie_bootstrap_sample(df, statistic, n_bootstrap = 1000L, seed = 42L)
```

Arguments

`df`	A data frame.
`statistic`	A function taking a data frame and returning a scalar.
`n_bootstrap`	Number of bootstrap replicates.
`seed`	Random seed.

Value

Named list: `estimate`, `se`, `ci_lower`, `ci_upper`, `distribution` (numeric vector of bootstrap statistics).

Examples

```
df <- data.frame(x = rnorm(100))
morie_bootstrap_sample(df, statistic = function(d) mean(d$x))
```

`morie_build_outputs_manifest`
Build an outputs manifest from a directory of artifacts

Description

Build an outputs manifest from a directory of artifacts

Usage

```
morie_build_outputs_manifest(
  output_dir,
  manifest_path,
  public_prefix = "data/manifest/outputs",
  extensions = c("csv", "pdf", "png", "html", "txt", "md")
)
```

Arguments

`output_dir`	Directory containing output files.
`manifest_path`	CSV path to write.
`public_prefix`	Prefix used in `public_path` values.
`extensions`	File extensions to include (without dots).

Value

Manifest data frame.

Examples

```
# Scan a tempdir of output files and build a manifest CSV:
tdir <- tempfile("morie-doc-")
dir.create(tdir)
writeLines("x,y\n1,2", file.path(tdir, "results.csv"))
writeLines("# report", file.path(tdir, "report.md"))
morie_build_outputs_manifest(tdir, file.path(tdir, "outputs_manifest.csv"))
```

`morie_builtin_db` *Get path to the built-in MORIE datasets database*

Description

Returns the path to `morie.db` that ships with the package (`inst/extdata/morie.db`). This database contains all CPADS, CCS, CSADS, CSUS, HealthInfobase, and CIHI datasets pre-loaded as SQLite tables.

Usage

```
morie_builtin_db()
```

Value

File path string.

Examples

```
morie_builtin_db()
```

morie_cache_clear *Clear morie's persistent cache directory*

Description

Removes files cached by morie under `tools::R_user_dir("morie", "cache")` (or `MORIE_CACHE_DIR` if set). morie's default behaviour writes caches to a session-scoped `tempdir()` subdirectory, so this function only matters if you have explicitly opted in to persistent caching by passing `cache_dir = morie_cache_dir(...)` to any of the morie fetchers.

Usage

```
morie_cache_clear(subdir = NULL, confirm = interactive())
```

Arguments

`subdir`	Optional subdirectory under the morie cache root to target (e.g. `"siu"`, `"tps"`). If `NULL`, removes the entire morie persistent-cache root.
`confirm`	If `TRUE` (default in interactive sessions), prompts the user before deleting. Set `FALSE` in scripts / batch use to skip the prompt.

Value

Invisibly, the number of files removed.

See Also

`morie_cache_dir`

Examples

```
# Non-interactive: skip the confirmation prompt.
morie_cache_clear("siu", confirm = FALSE)
```

morie_cache_dir *morie cache contract*

Description

morie functions that persist artifacts to disk (e.g. `morie_fetch_siu(cache_html = TRUE)`) default to a *session-scoped* subdirectory of `tempdir()`, which R automatically removes when the session ends. This is the most conservative CRAN-Policy-compliant default: nothing morie writes ever survives the R session unless the user explicitly opts in.

Usage

```
morie_cache_dir(subdir = NULL)
```

Arguments

`subdir`	Optional subdirectory under the morie cache root (e.g. `"siu"`, `"tps"`). If `NULL`, the cache root itself is returned.

Details

Users who want *persistent* caching across sessions opt in by passing the result of `morie_cache_dir(subdir)` as the `cache_dir` argument, e.g.:

```
morie_fetch_siu(
  cache_dir = morie_cache_dir("siu"),
  cache_html = TRUE
)
```

The persistent location is `tools::R_user_dir("morie", "cache")` (R $\geq$ 4.0), which on Linux defaults to `~/.cache/R/morie/`, on macOS to `~/Library/Caches/org.R-project.R/R/morie/`, and on Windows to `%LOCALAPPDATA%/R/cache/R/morie/`. Users can override this location by setting the `MORIE_CACHE_DIR` environment variable before calling `morie_cache_dir()`.

Active management. CRAN Policy requires persistent caches to be actively managed. Use `morie_cache_clear()` to empty the persistent cache (or a subdirectory of it). Cached SIU HTML is ~80-100 MB at full sweep, so clearing it occasionally is usually unnecessary, but it is supported.

Value

A file path string. The directory is *not* created; callers create it lazily only when they actually persist to disk.

See Also

`morie_cache_clear`

Examples

```
# Persistent cache root (does not write anything to disk):
morie_cache_dir()
# Per-subsystem persistent path:
morie_cache_dir("siu")
```

morie_cache_file *Cache local RDS/CSV data into the SQLite database*

Description

Reads a local file and writes it to the cache so that CI and Docker environments (which may lack the original files) can still run tests.

Usage

```
morie_cache_file(path, table_name, db_path = NULL, con = NULL)
```

Arguments

path	Path to a CSV or RDS file.
table_name	Name for the cached table.
db_path	Optional path to a SQLite file (default backend).
con	Optional pre-opened DBI connection (overrides db_path).

Value

Number of rows cached (invisible).

Examples

```
tdir <- tempfile("morie-cache-")
dir.create(tdir)
f <- file.path(tdir, "demo.csv")
write.csv(data.frame(x = 1:3, y = 4:6), f, row.names = FALSE)
morie_cache_file(f, "demo", db_path = file.path(tdir, "cache.db"))
```

morie_cache_list *List all tables in the MORIE cache*

Description

List all tables in the MORIE cache

Usage

```
morie_cache_list(db_path = NULL, con = NULL)
```

Arguments

db_path	Optional path to a SQLite file (default backend).
con	Optional pre-opened DBI connection (overrides db_path).

Value

A data.frame with columns `table` and `rows`.

Examples

```
db <- tempfile(fileext = ".db")
morie_cache_store(data.frame(x = 1:3), "demo", db_path = db)
morie_cache_list(db_path = db)
file.remove(db)
```

`morie_cache_load` *Load a table from the MORIE cache*

Description

Load a table from the MORIE cache

Usage

```
morie_cache_load(table_name, db_path = NULL, con = NULL)
```

Arguments

`table_name`	Name of the table.
`db_path`	Optional path to a SQLite file (default backend).
`con`	Optional pre-opened DBI connection (overrides `db_path`).

Value

A data.frame, or `NULL` if the table does not exist.

Examples

```
db <- tempfile(fileext = ".db")
morie_cache_store(
  data = data.frame(x = 1:5),
  table_name = "demo",
  db_path = db
)
morie_cache_load(table_name = "demo", db_path = db)
file.remove(db)
```

morie_cache_store *Store a data frame in the MORIE cache*

Description

Writes (or replaces) a table in the shared SQLite cache.

Usage

```
morie_cache_store(data, table_name, db_path = NULL, con = NULL)
```

Arguments

data	A data.frame to cache.
table_name	Name of the destination table.
db_path	Optional path to a SQLite file (default backend).
con	Optional pre-opened DBI connection. When supplied, the table is written through `con` and `db_path` is ignored. Use this for non-SQLite backends (PostgreSQL, DuckDB, MariaDB).

Value

Number of rows written (invisible).

Examples

```
db <- tempfile(fileext = ".db")
morie_cache_store(
  data = data.frame(x = rnorm(50), y = rnorm(50)),
  table_name = "demo",
  db_path = db
)
file.remove(db)
```

morie_calculate_ebac *Calculate estimated Blood Alcohol Concentration (eBAC)*

Description

Compute the continuous estimated Blood Alcohol Concentration using the standard Widmark formula. Mirrors the Python `morie.calculate_ebac()`.

Usage

```
morie_calculate_ebac(drinks, weight_lbs, hours, gender_constant)
```

Arguments

`drinks`	Number of standard drinks consumed (1 drink = 14 g alcohol).
`weight_lbs`	Body weight in pounds.
`hours`	Hours elapsed since drinking began.
`gender_constant`	Widmark gender multiplier (0.73 men, 0.66 women).

Details

The Widmark formula is:

$$eBAC = (drinks \times 5.14)/(weight_lbs \times r) - 0.015 \times hours$$

where r is the gender constant (0.73 for men, 0.66 for women). Returned values are clipped at zero.

Value

Non-negative numeric scalar: estimated BAC.

Examples

```
morie_calculate_ebac(drinks = 4, weight_lbs = 180, hours = 2, gender_constant = 0.73)
```

`morie_calculate_ipw_weights` *Calculate inverse probability of treatment weights (IPTW)*

Description

Mirrors the Python `morie.calculate_ipw_weights()`. Pure-R, no extra dependencies.

Usage

```
morie_calculate_ipw_weights(
  data,
  treatment,
  ps_col,
  stabilized = FALSE,
  trim_quantiles = NULL
)
```

Arguments

`data`	A `data.frame` containing treatment assignment and propensity scores.
`treatment`	Column name (string) of the binary treatment.
`ps_col`	Column name (string) of the propensity scores.
`stabilized`	If `TRUE`, return stabilised IPW weights. Default `FALSE`.
`trim_quantiles`	Optional length-2 numeric vector (q_l, q_u) in ‘$[0, 1]$‘; if supplied, weights are clipped to the q_l-th and q_u-th quantiles of the unclipped weight distribution (Crump et al. 2009 trimming). Default `NULL`.

Details

Standard IPW: $w_i = T_i/e_i + (1 - T_i)/(1 - e_i)$, with the propensity score e_i clipped at [0.01, 0.99] for stability. Stabilised IPW replaces T and $1 - T$ with the marginal treatment probability $P(T = 1)$ and $P(T = 0)$ respectively.

Value

Numeric vector of IPTW weights, length `nrow(data)`.

Examples

```
set.seed(1)
df <- data.frame(
  t = rbinom(100, 1, 0.4),
  ps = pmin(pmax(runif(100, 0.05, 0.95), 0.05), 0.95)
)
w <- morie_calculate_ipw_weights(df, treatment = "t", ps_col = "ps")
summary(w)
```

`morie_calibration_weights` *Calibration weights via iterative proportional fitting (raking)*

Description

Adjusts initial design weights so that weighted marginal totals match known population totals for each auxiliary variable.

Usage

```
morie_calibration_weights(
  df,
  aux_vars,
  population_totals,
  initial_weights = NULL,
  max_iter = 50L,
  tol = 1e-06
)
```

Arguments

`df`	A data frame.
`aux_vars`	Character vector of categorical auxiliary variable names.
`population_totals`	Named list: `"var_level"` -> population count. Keys should be `"varname_level"` (e.g. `"gender_female"`).
`initial_weights`	Optional numeric vector of starting weights.
`max_iter`	Maximum IPF iterations.
`tol`	Convergence tolerance.

Value

Numeric vector of calibrated weights.

Examples

```
set.seed(1)
df <- data.frame(
  region = sample(c("A", "B"), 100, TRUE),
  sex = sample(c("M", "F"), 100, TRUE)
)
totals <- list(region_A = 60, region_B = 40, sex_M = 55, sex_F = 45)
morie_calibration_weights(df,
  aux_vars = c("region", "sex"),
  population_totals = totals
)
```

`morie_canonicalize_cpads_data` *Canonicalize raw CPADS PUMF columns*

Description

Canonicalize raw CPADS PUMF columns

Usage

```
morie_canonicalize_cpads_data(data)
```

Arguments

`data`	Raw CPADS data frame.

Value

Data frame with canonical MORIE analysis columns.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_causal_impact` *Bayesian structural time-series intervention analysis*

Description

Thin wrapper around `CausalImpact::CausalImpact()` (Brodersen et al. 2015). Fits a Bayesian structural time-series counterfactual to a single-series treatment using the pre-intervention window and reports the post-intervention causal effect with credible intervals.

Usage

```
morie_causal_impact(
  data,
  pre_period,
  post_period,
  model_args = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A data frame, matrix, or `zoo` object whose first column is the outcome and remaining columns are concurrent covariate predictors.
`pre_period`	Integer length-2 vector giving the start and end row indices (or time indices for `zoo`) of the pre-intervention window.
`post_period`	Integer length-2 vector giving the start and end row indices of the post-intervention window.
`model_args`	Optional named list passed to `CausalImpact::CausalImpact()`'s `model.args` argument (e.g. `list(niter = 1000L)`).
`alpha`	Posterior credible-interval coverage (default 0.05, meaning 95 percent intervals).

Details

Hard-errors if **CausalImpact** is not installed – the upstream Kalman-filter + slab-and-spike machinery has no compact inline equivalent. The wrapper is documented as an extender so that downstream rmorie callers have a stable `morie_*` entry point to the package.

Value

Named list with elements `average_effect`, `cumulative_effect`, `ci_lower`, `ci_upper`, `posterior_prob_causal`, and `summary` (the upstream `CausalImpact` summary matrix), plus the original `impact` object.

References

Brodersen KH, Gallusser F, Koehler J, Remy N, Scott SL (2015). Inferring causal impact using Bayesian structural time-series models. *Annals of Applied Statistics*, 9(1):247-274.

`morie_causal_robust_se`

Robust / heteroskedasticity-consistent variance for a fitted model

---

Description

Thin wrapper around **sandwich** variance estimators. Returns the requested variance-covariance matrix and the corresponding robust standard errors. Supports HC0-HC5 (cross-section), HAC (time-series), and clustered (one-way) variance.

Usage

```
morie_causal_robust_se(model, type = "HC3", cluster = NULL, ...)
```

Arguments

`model`	A fitted model object (typically from `stats::lm` or `stats::glm`) compatible with `sandwich::vcovHC()`, `sandwich::vcovHAC()`, or `sandwich::vcovCL()`.
`type`	One of `"HC0"`, `"HC1"`, `"HC2"`, `"HC3"`, `"HC4"`, `"HC4m"`, `"HC5"`, `"HAC"`, or `"CL"` (clustered). Default `"HC3"` (Long-Ervin small-sample default).
`cluster`	Optional one-sided formula or vector identifying the cluster variable (required when `type = "CL"`).
`...`	Additional arguments forwarded to the chosen sandwich estimator.

Details

Hard-errors if **sandwich** is not installed – the HC sandwich algebra is well-tested upstream and re-implementing it inline would be both lengthy and error-prone.

Value

Named list with elements `vcov` (variance matrix), `se` (named numeric vector of robust SEs), `type`, and `n_coef`.

References

Zeileis A, Koll S, Graham N (2020). Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R. *Journal of Statistical Software*, 95(1), 1-36.

`morie_causal_weighting` *Estimate balancing weights via* **WeightIt**

Description

Thin wrapper around `WeightIt::weightit()` exposing the full WeightIt method palette (`"glm"`, `"cbps"`, `"ebal"`, `"ps"`, `"energy"`, `"optweight"`, and any future additions). Provides MORIE callers with a stable `morie_*` entry point for balancing weights while preserving the underlying object so callers can pipe into `survey::svyglm` or `cobalt::bal.tab` downstream.

Usage

```
morie_causal_weighting(
  data,
  treatment,
  covariates,
  method = "glm",
  estimand = c("ATE", "ATT", "ATC"),
  ...
)
```

Arguments

`data`	A data frame.
`treatment`	Name of the treatment column (binary, multinomial, or continuous depending on `method`).
`covariates`	Character vector of covariate names.
`method`	One of `"glm"`, `"cbps"`, `"ebal"`, `"ps"`, `"energy"`, `"optweight"`, or any other method accepted by `WeightIt::weightit()`.
`estimand`	One of `"ATE"`, `"ATT"`, `"ATC"`; defaults to `"ATE"`.
`...`	Additional arguments forwarded to `WeightIt::weightit()`.

Details

Hard-errors if **WeightIt** is not installed – the multi-method weighting machinery has no compact inline equivalent.

Value

Named list with elements `weights` (numeric vector), `propensity_scores` (numeric vector or `NULL`), `method`, `estimand`, `ess` (effective sample size), and `weightit` (the original WeightIt object).

References

Greifer N (2024). WeightIt: Weighting for Covariate Balance in Observational Studies. R package version 1.4.0.

`morie_check_plugin_license` *Check whether a downstream package's SPDX is GPL-compatible*

Description

Check whether a downstream package's SPDX is GPL-compatible

Usage

```
morie_check_plugin_license(plugin_spdx, raise_on_incompatible = FALSE)
```

Arguments

`plugin_spdx`	SPDX identifier (e.g. `"MIT"`, `"Apache-2.0"`).
`raise_on_incompatible`	If `TRUE`, throw an error rather than warning when the licence is not GPL-compatible.

Value

`TRUE` if compatible. Issues a warning (or error) otherwise.

Examples

```
morie_check_plugin_license("MIT")
## Not run:
# The next call demonstrates the error path; runs only on
# explicit example() with run.dontrun = TRUE.
morie_check_plugin_license("LicenseRef-Proprietary",
  raise_on_incompatible = TRUE
)

## End(Not run)
```

morie_chi_square_test

Chi-square test of independence or goodness-of-fit

Description

Chi-square test of independence or goodness-of-fit

Usage

```
morie_chi_square_test(observed, expected = NULL)
```

Arguments

observed	Observed counts (matrix for independence, vector for GOF).
expected	Expected counts for GOF (optional; uniform if NULL).

Value

Named list: chi_sq, df, p_value, cramers_v.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_ckan_search

Search any CKAN open-data portal for datasets

Description

Wraps the CKAN package_search action so users can discover datasets that are not in the built-in MORIE catalog and fetch them through morie_fetch_ckan or morie_fetch.

Usage

```
morie_ckan_search(query, portal = "open.canada.ca", rows = 25L, ...)
```

Arguments

query	Free-text search string.
portal	A known portal name ("open.canada.ca", "data.ontario.ca", "open.toronto.ca") or a full CKAN base URL (e.g. "https://catalogue.example.org").
rows	Maximum number of datasets to return (default 25).
...	Extra named CKAN package_search parameters (e.g. fq = "res_format:CSV", sort = "metadata_modified desc").

Value

A data.frame with one row per resource, columns: `dataset_title`, `dataset_id`, `resource_id`, `resource_name`, `format`, `datastore_active`, `url`. Feed `resource_id` into `morie_fetch_ckan(resource_id = ...)`.

See Also

`morie_fetch_ckan`, `morie_fetch`

Examples

```
## Not run:
hits <- morie_ckan_search("cannabis survey", portal = "open.canada.ca")
head(hits[, c("dataset_title", "resource_id", "format")])

## End(Not run)
```

---

morie_classify_variable

Classify one variable.

Description

Classify one variable.

Usage

```
morie_classify_variable(
  col_name,
  dtype = "string",
  valid_values = NULL,
  dataset_name = "unknown"
)
```

Arguments

`col_name`	Character; the column name.
`dtype`	Character; one of `int` / `float` / `string` / `date` / `datetime` / `bool`.
`valid_values`	Optional character vector of closed-set values.
`dataset_name`	Character; the owning dataset id (e.g. `"b01"` for OTIS, `"uof_main_records"` for ARSAU).

Value

A named list with classes `morie_variable_taxonomy` / `list`.

`morie_cli_dump_catalog` *Emit a unified catalog CSV for the rmorie-cli binary*

Description

Walks `morie_datasets_browse()` (9242 datasets across all portals) and writes a single CSV with the columns the C++ `rmorie` binary's catalog parser expects: `dataset_key`, `portal`, `title`, `description`, `url`, `license`, `formats`.

Usage

```
morie_cli_dump_catalog(out_path = NULL)
```

Arguments

`out_path`	Where to write. Defaults to `inst/extdata/_unified_catalog.csv` inside the installed package (or the dev tree if running under `devtools::load_all()`).

Details

The CLI binary prefers this unified file when present; otherwise it falls back to scanning every `inst/extdata/*_catalog.csv` (which have heterogeneous per-portal schemas and don't parse cleanly).

Idempotent: regenerate after any change to the catalog registry.

Value

Invisibly, the path written.

`morie_cluster_sample` *Two-stage cluster sampling*

Description

Randomly selects `n_clusters` clusters, then takes all units within selected clusters.

Usage

```
morie_cluster_sample(df, cluster_col, n_clusters, seed = 42L)
```

Arguments

`df`	A data frame.
`cluster_col`	Name of the cluster identifier column.
`n_clusters`	Number of clusters to select.
`seed`	Random seed.

Value

Data frame of selected units with `.weight` column.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_cnn_genomic` *CNN genomic predictor (Conv1D + GAP + dense, base R)*

Description

CNN genomic predictor (Conv1D + GAP + dense, base R)

Usage

```
morie_cnn_genomic(
  x,
  y,
  markers,
  n_filters = 8,
  kernel = 3,
  hidden = 8,
  n_epochs = 150,
  lr = 0.01,
  l2 = 0.001,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	Optional fixed-effect design.
`y`	Numeric response.
`markers`	(n x m) genotype matrix.
`n_filters`, `kernel`, `hidden`, `n_epochs`, `lr`, `l2`, `seed`	Hyperparameters.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("cnnge", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, y_hat, W_conv, b_conv, W1, b1, w2, b2, se, n, method).

References

Montesinos Lopez Ch 13.

Examples

```
morie_cnn_genomic(x = rnorm(50), y = rnorm(50), markers = matrix(sample(0:2, 200, TRUE), 50, 4))
```

---

morie_coherence *Magnitude-squared morie_coherence between two time series*

Description

Magnitude-squared morie_coherence between two time series

Usage

```
morie_coherence(x, y, nperseg = NULL, fs = 1)
```

Arguments

x	Numeric vector.
y	Numeric vector (same length).
nperseg	Segment length. Default n/4.
fs	Sampling frequency. Default 1.

Value

Named list with `frequencies`, `morie_coherence`, `n_segments`, `nperseg`, `fs`, `n`, `method`.

Examples

```
morie_coherence(x = rnorm(50), y = rnorm(50))
```

---

morie_coin_independence

General independence test via **coin**

Description

Thin extender over `coin::independence_test` for conditional / permutation tests of independence between arbitrary response and covariate combinations.

Usage

```
morie_coin_independence(formula, data, ...)
```

Arguments

`formula`	A model formula, e.g. `y ~ x | block`.
`data`	A data frame.
`...`	Further arguments forwarded to `coin::independence_test` (e.g. `distribution`, `teststat`, `ytrafo`, `xtrafo`).

Value

A list with `$method = "coin::independence_test"` and `$raw` (an `IndependenceTest` object).

`morie_coin_oneway` *Permutation one-way ANOVA via* **coin**

Description

Thin extender over `coin::oneway_test` for the permutation analogue of the classical one-way ANOVA.

Usage

```
morie_coin_oneway(formula, data, ...)
```

Arguments

`formula`	A formula `y ~ group`, where `group` is a (multi-level) factor.
`data`	A data frame.
`...`	Further arguments forwarded to `coin::oneway_test` (e.g. `distribution`, `teststat`).

Value

A list with `$method = "coin::oneway_test"` and `$raw` (an `IndependenceTest` object).

morie_coin_wilcoxon *Permutation Wilcoxon test via* **coin**

Description

Thin extender over `coin::wilcox_test` for two-sample permutation Wilcoxon (Mann-Whitney) tests.

Usage

```
morie_coin_wilcoxon(formula, data, ...)
```

Arguments

`formula`	A two-sided formula `y ~ group`, where `group` is a two-level factor.
`data`	A data frame.
`...`	Further arguments forwarded to `coin::wilcox_test` (e.g. `distribution`, `alternative`).

Value

A list with `$method = "coin::wilcox_test"` and `$raw` (an `IndependenceTest` object).

morie_compare_nested_logistic_models *Compare nested logistic-regression models via likelihood-ratio test*

Description

Mirrors the Python `morie.compare_nested_logistic_models()`. Fits a reduced and a full logistic model (the reduced model's predictors must be a subset of the full model's), then performs an analysis-of-deviance LRT.

Usage

```
morie_compare_nested_logistic_models(
  data,
  outcome,
  predictors_full,
  predictors_reduced
)
```

Arguments

`data`	A `data.frame`.
`outcome`	Column name of the binary outcome.
`predictors_full`	Character vector: full model's predictors.
`predictors_reduced`	Character vector: reduced model's predictors. Must be a subset of `predictors_full`.

Value

A list with `chi_sq`, `df`, `p_value`, `aic_full`, `aic_reduced`, `n`.

Examples

```
set.seed(1)
df <- data.frame(
  y = rbinom(200, 1, 0.4),
  x1 = rnorm(200), x2 = rnorm(200), x3 = rnorm(200)
)
morie_compare_nested_logistic_models(df,
  outcome = "y",
  predictors_full = c("x1", "x2", "x3"),
  predictors_reduced = c("x1")
)
```

`morie_compute_design_weights` *Compute inverse-probability design weights*

Description

Compute inverse-probability design weights

Usage

```
morie_compute_design_weights(df, strata_col, population_sizes)
```

Arguments

`df`	A data frame.
`strata_col`	Name of the stratification column.
`population_sizes`	Named integer vector: stratum level -> population size.

Value

Numeric vector of design weights (same length as `nrow(df)`).

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_concordance_incomplete`
Kendall's coefficient of concordance W (Gibbons Ch 12.5)

Description

Supports incomplete rankings via NA entries. For complete rankings, W = 12 S / (k^2 (n^3 - n)) where S is the sum of squared deviations of object rank-sums from their mean. Significance via chi-square approximation k(n-1) W ~ chi-square with n-1 df.

Usage

```
morie_concordance_incomplete(x)
```

Arguments

`x`	Matrix (n objects rows x k rankers cols); NA = not ranked.

Value

Named list: statistic (W), p_value, df, chi2, n, k.

Examples

```
morie_concordance_incomplete(x = rnorm(50))
```

`morie_confusion_matrix_metrics`
Confusion matrix with precision / recall / F1 (R parity)

Description

Manually constructs the confusion matrix to avoid the caret dependency for what is fundamentally a tabulation.

Usage

```
morie_confusion_matrix_metrics(y_true, y_pred, labels = NULL)
```

Arguments

y_true	Observed labels.
y_pred	Predicted labels.
labels	Optional ordering vector.

Value

Named list: estimate (accuracy), accuracy, confusion_matrix, labels, precision, recall, f1, macro_precision, macro_recall, macro_f1, weighted_f1, n, method.

Examples

```
morie_confusion_matrix_metrics(y_true = rbinom(50, 1, 0.5), y_pred = rbinom(50, 1, 0.5))
```

---

morie_contingency_coefficient

Pearson contingency coefficient C (Gibbons Ch 14.2.1)

Description

C = sqrt(chi^2 / (chi^2 + n)). Also reports Cramer's V and the maximum attainable C = sqrt((min(r,c)-1)/min(r,c)).

Usage

```
morie_contingency_coefficient(x)
```

Arguments

x	A 2-D contingency table of counts.

Value

Named list: statistic (C), morie_cramers_v, chi2, p_value, df, max_C, n.

Examples

```
morie_contingency_coefficient(x = matrix(sample(1:5, 50, TRUE), 10, 5))
```

morie_control_comparison

Nonparametric many-to-one comparisons to a control (Gibbons Ch 10.7)

Description

Mann-Whitney vs. control for each treatment group; Bonferroni- adjusted p-values by default.

Usage

```
morie_control_comparison(
  groups,
  control_index = 1L,
  adjust = c("bonferroni", "none")
)
```

Arguments

`groups`	List of numeric vectors; first (or `control_index`-th) element is the control.
`control_index`	Integer position of the control group. Default 1.
`adjust`	One of `"bonferroni"`, `"none"`.

Value

Named list: statistic, p_value, p_adjusted, n, k, control_n.

Examples

```
morie_control_comparison(groups = list(rnorm(20), rnorm(20), rnorm(20)))
```

morie_control_median_test

Mood's median (control-median) test (Gibbons Ch 6.5)

Description

Two-sample median test: contingency-table chi-square on the counts above/below the pooled-sample median.

Usage

```
morie_control_median_test(x, y)
```

Arguments

x	Numeric vector (control).
y	Numeric vector (treatment).

Value

Named list: statistic, p_value, df, n, grand_median, table.

Examples

```
morie_control_median_test(x = rnorm(50), y = rnorm(50))
```

---

morie_copula_fit *Fit a copula by maximum-likelihood via* **copula**

Description

Thin extender over `copula::fitCopula` that estimates copula parameters from pseudo-observations on $[0,1]^d$.

Usage

```
morie_copula_fit(copula, data, ...)
```

Arguments

copula	A `copula` object specifying the parametric family (e.g. `copula::normalCopula()`, `copula::claytonCopula()`).
data	Numeric matrix of pseudo-observations on $[0,1]^d$, with one column per margin (typically obtained via `copula::pobs`).
...	Further arguments forwarded to `copula::fitCopula` (e.g. `method`, `start`, `optim.method`, `estimate.variance`).

Value

A list with `$method = "copula::fitCopula"` and `$raw` (a `fitCopula` object with the estimated parameters, log-likelihood and variance estimates).

Examples

```
## Not run:
  if (requireNamespace("copula", quietly = TRUE)) {
    set.seed(1)
    cop <- copula::normalCopula(0.5, dim = 2)
    u <- copula::rCopula(200, cop)
    morie_copula_fit(copula::normalCopula(dim = 2), data = u)
  }

## End(Not run)
```

`morie_copula_sample` *Draw a random sample from a copula via* **copula**

Description

Thin extender over `copula::rCopula` that generates n draws on $[0,1]^d$ from a specified copula.

Usage

```
morie_copula_sample(n, copula, ...)
```

Arguments

`n`	Integer; the number of multivariate observations to draw.
`copula`	A `copula` object specifying the dependence structure (e.g. `copula::normalCopula()`, `copula::claytonCopula()`).
`...`	Further arguments forwarded to `copula::rCopula`.

Value

A list with `$method = "copula::rCopula"` and `$raw` (a numeric matrix of dimension $n \times d$ with values in $[0,1]$).

Examples

```
## Not run:
  if (requireNamespace("copula", quietly = TRUE)) {
    set.seed(1)
    morie_copula_sample(100, copula::claytonCopula(2, dim = 3))
  }

## End(Not run)
```

`morie_corrections_uof_resource_ids` *Canonical CKAN resource id table for the Corrections UoF dataset.*

Description

Returns the 12 morie short-name -> CKAN resource-id map for the Ontario "Use of Force in Correctional Institutions" dataset on data.ontario.ca. Useful for catalog discovery + sanity tests.

Usage

```
morie_corrections_uof_resource_ids()
```

Value

Named list of 12 CKAN resource ids.

Examples

```
ids <- morie_corrections_uof_resource_ids()
length(ids)
names(ids)
```

---

morie_cpads_canonicalize_frame

Canonicalize raw CPADS PUMF columns into morie's analysis schema.

Description

First-pass canonicalization layer based on the public CPADS PUMF field names. If `frame` already carries the canonical columns it is returned unchanged (after validation). Otherwise raw PUMF columns are remapped using `.MORIE_CPADS_RAW_COLUMN_MAP` and missing/DKNR codes (98, 99) are converted to `NA`.

Usage

```
morie_cpads_canonicalize_frame(frame)
```

Arguments

`frame`	A `data.frame` carrying raw CPADS PUMF columns (or the already-canonical analysis columns).

Value

A `data.frame` with the canonical CPADS analysis columns.

`morie_cpads_contract` *Return the CPADS analysis-frame contract.*

Description

Describes the morie CPADS contract: the canonical analysis variables expected in a wrangled frame, the raw -> canonical column map, and the conventional on-disk cache path used when a user has wrangled the PUMF themselves.

Usage

```
morie_cpads_contract()
```

Details

CPADS is open data (Open Government Licence – Canada). The Public Use Microdata File is available at open.canada.ca, dataset `736fa9b2-62e4-4e31-aea4-51869605b363` (resource `d2639429-c304-45a6-90b3-770562f4d46d`, file `cpads-2021-2022-pumf2.csv`). Aggregate dashboards at https://health-infobase.canada.ca/substance-use/reports/cpads/. morie ships a 30-row synthetic at `inst/extdata/cpads_pumf_synthetic.csv` for offline CRAN-safe tests; `morie_datasets_cpads(offline = FALSE)` fetches the live PUMF. Earlier morie versions wrongly claimed CPADS was "FOI/agreement-only"; that was incorrect and has been retracted as of 3MMM.

Value

A named list with fields `source_kind`, `expected_wrangled_path`, `required_variables`, `raw_column_map`, and `note`.

Examples

```
contract <- morie_cpads_contract()
contract$required_variables
```

`morie_cpads_has_raw_columns` *Detect whether a data frame looks like raw CPADS PUMF data.*

Description

Accepts either `wtpumf` (PUMF release) or `wtdf` (full dataset) as the weight column; otherwise all documented raw PUMF columns must be present.

Usage

```
morie_cpads_has_raw_columns(frame)
```

Arguments

`frame`	A `data.frame`.

Value

Logical scalar; `TRUE` if the frame contains the raw CPADS PUMF schema, `FALSE` otherwise.

`morie_cpads_infer_file_format` *Infer the on-disk file format for a CPADS file path.*

Description

Recognises `.csv`, `.xlsx` / `.xls`, and `.rds`. Raises for any other extension.

Usage

```
morie_cpads_infer_file_format(path)
```

Arguments

`path`	A character scalar file path.

Value

One of `"csv"`, `"excel"`, or `"rds"`.

Examples

```
morie_cpads_infer_file_format("data/cache/cpads_pumf_wrangled.rds")
```

`morie_cpads_missing_variables` *Identify missing canonical CPADS variables in a column set.*

Description

Identify missing canonical CPADS variables in a column set.

Usage

```
morie_cpads_missing_variables(columns)
```

Arguments

`columns`	Character vector of column names (e.g. `colnames(df)`).

Value

Character vector of missing canonical CPADS variables (empty if every required variable is present).

Examples

```
morie_cpads_missing_variables(c("weight", "alcohol_past12m"))
```

`morie_cpads_validate_frame` *Validate a data frame against the canonical CPADS analysis contract.*

Description

Validate a data frame against the canonical CPADS analysis contract.

Usage

```
morie_cpads_validate_frame(frame, strict = TRUE)
```

Arguments

`frame`	A `data.frame` (or `tibble`).
`strict`	Logical; if `TRUE` (default), raise an error when any required variable is missing. If `FALSE`, return the missing names without raising.

Value

Character vector of missing canonical variable names (invisibly when strict and complete).

Examples

```
## Not run:
morie_cpads_validate_frame(df, strict = TRUE)

## End(Not run)
```

`morie_crypto_chacha20_poly1305_decrypt`
ChaCha20-Poly1305 IETF authenticated decryption

Description

Phase 3JJJ1. Inverse of `morie_crypto_chacha20_poly1305_encrypt()`. Accepts the full ciphertext || tag buffer (concatenate as `c(ct, tag)`).

Usage

```
morie_crypto_chacha20_poly1305_decrypt(key, nonce, ct_with_tag, aad = raw(0))
```

Arguments

`key`	32-byte raw vector.
`nonce`	12-byte raw vector.
`ct_with_tag`	Raw vector containing ciphertext appended with the 16-byte tag.
`aad`	Optional raw vector of additional authenticated data.

Value

Decrypted plaintext as raw vector.

`morie_crypto_chacha20_poly1305_encrypt`
ChaCha20-Poly1305 IETF authenticated encryption

Description

Phase 3JJJ1. Wraps libsodium's `crypto_aead_chacha20poly1305_ietf_encrypt` (RFC 8439 IETF variant: 32-byte key, 12-byte nonce, 16-byte authentication tag).

Usage

```
morie_crypto_chacha20_poly1305_encrypt(key, nonce, plaintext, aad = raw(0))
```

Arguments

`key`	32-byte raw vector.
`nonce`	12-byte raw vector (single-use per key; reuse is catastrophic).
`plaintext`	Raw vector to encrypt (may be empty).
`aad`	Optional raw vector of additional authenticated data (default empty).

Details

Byte-compatible with the Python morie `chacha20_poly1305_encrypt(key, nonce, plaintext, aad)`. The C transport returns ciphertext || tag as a single buffer; this R wrapper splits it into `list(ct = ..., tag = ...)` to match the Python tuple return shape.

Value

List with `ct` (raw vector, length = `length(plaintext)`) and `tag` (raw vector, 16 bytes).

Examples

```
if (morie_crypto_sodium_available()) {
  k <- morie_crypto_random_bytes(32)
  n <- morie_crypto_random_bytes(12)
  r <- morie_crypto_chacha20_poly1305_encrypt(k, n, charToRaw("hello"))
  p <- morie_crypto_chacha20_poly1305_decrypt(k, n, c(r$ct, r$tag))
  rawToChar(p)
}
```

`morie_crypto_hkdf_sha256` *HKDF-SHA256 (RFC 5869)*

Description

Phase 3JJJ1. Mirrors the Python `morie.crypto.hkdf_sha256(ikm, length=32, salt=b"", info=b"")` byte-for-byte. Empty `salt` defaults to a 32-byte zero-filled salt per RFC 5869 §2.2 (matches Python).

Usage

```
morie_crypto_hkdf_sha256(ikm, length = 32L, salt = raw(0), info = raw(0))

morie_crypto_hkdf_sha256(ikm, length = 32L, salt = raw(0), info = raw(0))
```

Arguments

`ikm`	Input keying material (raw vector).
`length`	Output length in bytes (1..8160).
`salt`	Optional salt raw vector. Empty -> zero-fill.
`info`	Optional context/application info raw vector.

Value

Raw vector of `length` bytes.

Derived key material as raw vector of length `length`.

`morie_crypto_hybrid_decrypt`
Hybrid decrypt: ML-KEM-768 + ChaCha20-Poly1305

Description

Hybrid decrypt: ML-KEM-768 + ChaCha20-Poly1305

Usage

```
morie_crypto_hybrid_decrypt(ciphertext, recipient_sk)
```

Arguments

`ciphertext`	Raw vector container.
`recipient_sk`	Raw vector: recipient's ML-KEM-768 secret key.

Value

Raw vector of decrypted plaintext.

`morie_crypto_hybrid_encrypt`
Hybrid encrypt: ML-KEM-768 + ChaCha20-Poly1305

Description

Hybrid encrypt: ML-KEM-768 + ChaCha20-Poly1305

Usage

```
morie_crypto_hybrid_encrypt(plaintext, recipient_pk)
```

Arguments

`plaintext`	Raw vector or character string to encrypt.
`recipient_pk`	Raw vector: recipient's ML-KEM-768 public key.

Value

Raw vector container.

`morie_crypto_hybrid_keygen` *Generate an ML-KEM-768 key pair for hybrid encryption*

Description

Generate an ML-KEM-768 key pair for hybrid encryption

Usage

```
morie_crypto_hybrid_keygen()
```

Value

A named list with `pk` (raw) and `sk` (raw).

`morie_crypto_keystore_create` *Create a new empty morie keystore*

Description

Create a new empty morie keystore

Usage

```
morie_crypto_keystore_create(password, path = .morie_keystore_default_path())
```

Arguments

`password`	Character scalar: keystore password.
`path`	File path.

Value

Invisibly, NULL.

`morie_crypto_keystore_list` *List key names in the morie keystore*

Description

List key names in the morie keystore

Usage

```
morie_crypto_keystore_list(password, path = .morie_keystore_default_path())
```

Arguments

`password`	Character scalar.
`path`	Keystore path.

Value

Character vector of identifiers.

`morie_crypto_keystore_load` *Load a key pair from the morie keystore*

Description

Load a key pair from the morie keystore

Usage

```
morie_crypto_keystore_load(
  name,
  password,
  path = .morie_keystore_default_path()
)
```

Arguments

`name`	Identifier.
`password`	Character scalar.
`path`	Keystore path.

Value

Named list with pk (raw) and sk (raw).

`morie_crypto_keystore_store` *Store a key pair in the morie keystore*

Description

Store a key pair in the morie keystore

Usage

```
morie_crypto_keystore_store(
  name,
  pk,
  sk,
  password,
  path = .morie_keystore_default_path()
)
```

Arguments

`name`	Identifier.
`pk`	Raw vector: public key.
`sk`	Raw vector: secret key.
`password`	Character scalar.
`path`	Keystore path.

Value

Invisibly, NULL.

`morie_crypto_liboqs_available` *Is liboqs available in this morie build?*

Description

Phase 3JJJ2. Returns `TRUE` when morie's compiled .so was linked against the Open Quantum Safe library at install time. If `FALSE`, install liboqs and reinstall morie:

Usage

```
morie_crypto_liboqs_available()
```

Details

- macOS: `brew install liboqs`
- Debian: `sudo apt-get install liboqs-dev` (or build from source: https://github.com/open-quantum-safe/liboqs)

Value

Single logical.

`morie_crypto_liboqs_version` *liboqs runtime version string*

Description

liboqs runtime version string

Usage

```
morie_crypto_liboqs_version()
```

Value

Single character (e.g. `"0.15.0"`); empty if liboqs absent.

`morie_crypto_mldsa65_keygen` *ML-DSA-65 keypair generation (NIST FIPS 204)*

Description

Phase 3JJJ2. Generates a post-quantum signature keypair. Sizes: `pk` = 1952 bytes, `sk` = 4032 bytes.

Usage

```
morie_crypto_mldsa65_keygen()
```

Value

List with `pk` (raw, 1952 B) and `sk` (raw, 4032 B).

`morie_crypto_mldsa65_sign` *ML-DSA-65 signature*

Description

Sign a message with an ML-DSA-65 secret key. Signature length is variable up to a 3309-byte ceiling (typical: ~3293 B).

Usage

```
morie_crypto_mldsa65_sign(sk, message)
```

Arguments

`sk`	4032-byte raw vector (signer's secret key).
`message`	Raw vector to sign.

Value

Raw vector signature.

`morie_crypto_mldsa65_verify` *ML-DSA-65 signature verification*

Description

ML-DSA-65 signature verification

Usage

```
morie_crypto_mldsa65_verify(pk, message, signature)
```

Arguments

`pk`	1952-byte raw vector (signer's public key).
`message`	Raw vector that was signed.
`signature`	Raw vector signature returned by `morie_crypto_mldsa65_sign()`.

Value

Single logical: `TRUE` if signature is valid.

`morie_crypto_mlkem768_decaps`
ML-KEM-768 decapsulation

Description

Recover the shared secret from an encapsulation ciphertext using the recipient's secret key.

Usage

```
morie_crypto_mlkem768_decaps(sk, ct)
```

Arguments

`sk`	2400-byte raw vector (recipient's ML-KEM-768 secret key).
`ct`	1088-byte raw vector (sender's encapsulation ciphertext).

Value

Raw vector (32 B), the shared secret.

`morie_crypto_mlkem768_encaps`
ML-KEM-768 encapsulation

Description

Encapsulate a shared secret under a recipient's ML-KEM-768 public key. Returns the ciphertext (1088 B) the sender transmits, plus the 32-byte shared secret the sender holds locally.

Usage

```
morie_crypto_mlkem768_encaps(pk)
```

Arguments

`pk`	1184-byte raw vector (recipient's ML-KEM-768 public key).

Value

List with `ct` (raw, 1088 B) and `shared_secret` (raw, 32 B).

`morie_crypto_mlkem768_keygen`
ML-KEM-768 keypair generation (NIST FIPS 203)

Description

Phase 3JJJ2. Generates a post-quantum key encapsulation keypair. Sizes: `pk` = 1184 bytes, `sk` = 2400 bytes.

Usage

```
morie_crypto_mlkem768_keygen()
```

Value

List with `pk` (raw, 1184 B) and `sk` (raw, 2400 B).

Examples

```
if (morie_crypto_liboqs_available()) {
  kp <- morie_crypto_mlkem768_keygen()
  c(pk = length(kp$pk), sk = length(kp$sk))
}
```

`morie_crypto_random_bytes`
Cryptographically secure random bytes (libsodium)

Description

Phase 3JJJ1. Wraps libsodium's `randombytes_buf`.

Usage

```
morie_crypto_random_bytes(n)
```

Arguments

`n`	Number of bytes to generate.

Value

Raw vector of length `n`.

`morie_crypto_sodium_available`
Is libsodium available in this morie build?

Description

Phase 3JJJ1. Returns `TRUE` when morie's compiled .so was linked against libsodium at install time (detected by ./configure via `pkg-config --libs libsodium` or a bare `-lsodium` probe). If `FALSE`, install libsodium and reinstall morie:

Usage

```
morie_crypto_sodium_available()
```

Details

- macOS: `brew install libsodium`
- Debian: `sudo apt-get install libsodium-dev`
- Fedora: `sudo dnf install libsodium-devel`

Value

Single logical.

Examples

```
morie_crypto_sodium_available()
```

`morie_crypto_sodium_version`
libsodium runtime version string

Description

Phase 3JJJ1. Returns the bundled libsodium version (e.g., `"1.0.20"`); empty string if libsodium wasn't linked.

Usage

```
morie_crypto_sodium_version()
```

Value

Single character.

`morie_dataset_catalog` *List all datasets in the MORIE catalog*

Description

Returns a data.frame describing every dataset available through the MORIE data management system. Each row maps a short catalog key to its source, survey, year, file format, local path, SQLite table name, and CKAN resource ID (if available).

Usage

```
morie_dataset_catalog()
```

Details

Keys match the Python DATASET_CATALOG in `data.py` exactly. Use `morie_load_dataset` to load by key.

Value

A data.frame with 44 rows (one per dataset) and columns: key, name, source, survey, year, format, type, large_file, local_path, table_name, ckan_resource_id, download_url, zip_member. The `download_url` / `zip_member` columns are empty for datasets reachable through the SQLite cache or the CKAN datastore.

Examples

```
cat <- morie_dataset_catalog()
nrow(cat)
head(cat[, c("key", "name", "source", "year")])
# Find Ontario carceral datasets:
cat[
  grepl("OTIS|Ontario", paste(cat$source, cat$survey)),
  c("key", "year")
]
```

`morie_dataset_column_profile` *Build a single-column profile record.*

Description

Build a single-column profile record.

Usage

```
morie_dataset_column_profile(
  series,
  name,
  ordinal_threshold = 10L,
  binary_threshold = 2L
)
```

Arguments

`series`	A vector.
`name`	Column name.
`ordinal_threshold`	Integer; passed to `morie_dataset_infer_level()`.
`binary_threshold`	Integer; max unique values to count as binary (default 2).

Value

Named list with fields `name`, `dtype`, `level`, `n_unique`, `missing_pct`, `is_binary`, `is_constant`, `suggested_role`, `summary_stats`.

`morie_dataset_detect_role` *Detect the suggested epidemiological role of a column.*

Description

Detect the suggested epidemiological role of a column.

Usage

```
morie_dataset_detect_role(x, name)
```

Arguments

`x`	A vector.
`name`	Column name (drives the heuristic patterns).

Value

One of `"id"`, `"weight"`, `"stratum"`, `"cluster"`, `"treatment"`, `"outcome"`, `"covariate"`.

`morie_dataset_infer_level` *Infer the Stevens NOIR measurement level for a single vector.*

Description

Decision rules, in order:

1. Character / factor with `n_unique <= ordinal_threshold` and an ordinal name hit (likert/grade/scale/...): `"ordinal"`.
2. Character / factor otherwise: `"nominal"`.
3. Logical: `"nominal"`.
4. Numeric with `n_unique <= 2` (binary): `"nominal"`.
5. Numeric with `n_unique <= 20` + ordinal name hit: `"ordinal"`.
6. Double with interval name hit (year/index/date/...): `"interval"`.
7. Double otherwise: `"ratio"`.
8. Integer with non-negative range: `"ratio"`; else `"interval"`.
9. Date / POSIXct: `"interval"`.

Usage

```
morie_dataset_infer_level(x, name = NULL, ordinal_threshold = 10L)
```

Arguments

`x`	A vector (any atomic type or factor).
`name`	Optional column name to drive the name-based heuristics. Defaults to `NULL` (no name-based promotion).
`ordinal_threshold`	Integer; max unique values for a categorical column to be considered ordinal (default 10).

Value

Character scalar; one of `"nominal"`, `"ordinal"`, `"interval"`, `"ratio"`.

morie_dataset_info *Get metadata for a single dataset*

Description

Get metadata for a single dataset

Usage

```
morie_dataset_info(key)
```

Arguments

key	Dataset catalog key (or fuzzy match).

Value

A named list with dataset metadata.

Examples

```
# Use a real catalog key (run `morie_dataset_catalog()$key` to list them):
info <- morie_dataset_info("ocp21")
info$source
info$year
# Fuzzy match works for partial / forgiving keys:
morie_dataset_info("cpads")$key
```

morie_dataset_load *Load a dataset from a CSV / TSV / Excel / Parquet / JSON file.*

Description

File format is detected from the extension. Supported extensions: `.csv`, `.tsv`, `.xlsx` / `.xls`, `.parquet` / `.pq`, `.json` / `.jsonl`.

Usage

```
morie_dataset_load(path, encoding = "UTF-8", ...)
```

Arguments

path	Character; file path.
encoding	Character; encoding for text formats (default `"UTF-8"`).
...	Forwarded to the underlying reader (`utils::read.csv`, `readxl::read_excel`, etc.).

Value

A `data.frame`.

`morie_dataset_portal_catalog` *Build the cross-portal dataset catalog*

Description

Aggregates every per-portal registry (Chicago, NYC NYPD, NYC OpenData, TPS ArcGIS Hub, TPS PSDP, Ontario CKAN, Vancouver, VPD GeoDASH, Statistics Canada CCJS, Montreal, Toronto, Calgary, Edmonton, Ottawa) into a single tidy `data.frame` for cross-portal discovery and tooling. Caches the result in a session-local environment so repeated calls are O(1); call `morie_dataset_portal_catalog_clear_cache()` to force a rebuild after editing a registry in an interactive session.

Usage

```
morie_dataset_portal_catalog(portal = NULL)
```

Arguments

`portal`	Optional character filter restricting output to a single portal. `NULL` (default) returns every dataset across all registries. Bulk portals (NYC OpenData, Chicago, Toronto Hub, etc.) prefer the `rmoriedata` companion when installed, otherwise contribute zero rows with a one-time warning per portal. A per-portal call works without `rmoriedata` for portals whose registry lives in code (`"nyc_nypd"`, `"tps_psdp"`, `"ontario_ckan"`, `"statcan_ccjs"`, etc.); portals served by `rmoriedata` (`"nyc_opendata"`, `"chicago"`, `"tps_arcgis_hub"`, `"vancouver_opendata"`, etc.) return zero rows with a one-time warning when the companion is absent.

Value

A `data.frame` with columns `dataset_key`, `source`, `id`, `api_modes`, `loader`, `dict_url`, `n_rows_bundled`.

See Also

`morie_dataset_portal_catalog_clear_cache()`, `morie_datasets_load_by_key()`, `morie_datasets_browse()`

Examples

```
# Per-portal slice: registry lives in code, fastest path.
nypd <- morie_dataset_portal_catalog(portal = "nyc_nypd")
nrow(nypd)
head(nypd$dataset_key)
```

```
# Full catalog: bulk portals (NYC OpenData, Chicago, Toronto Hub,
# etc.) prefer the rmoriedata companion when installed, otherwise
# contribute zero rows with a one-time warning per portal.
cat_df <- morie_dataset_portal_catalog()
table(cat_df$source)
```

`morie_dataset_portal_catalog_clear_cache` *Clear the session-scoped portal-catalog cache*

Description

Forces the next `morie_dataset_portal_catalog()` call to rebuild from the per-portal registries. Useful after editing or extending a registry in an interactive session.

Usage

```
morie_dataset_portal_catalog_clear_cache()
```

Value

Invisibly `NULL`.

`morie_dataset_profile` *Fully profile a data frame without prior schema knowledge.*

Description

Walks every column, infers its NOIR level and epidemiological role, computes summary statistics, and resolves a best-guess treatment, outcome, and survey-weight column. User-supplied hints override heuristic detection.

Usage

```
morie_dataset_profile(
  df,
  hint_treatment = NULL,
  hint_outcome = NULL,
  hint_weights = NULL,
  ordinal_threshold = 10L,
  binary_threshold = 2L
)
```

Arguments

df	A data.frame.
hint_treatment	Optional character; force this column as the treatment.
hint_outcome	Optional character; force this column as the outcome.
hint_weights	Optional character; force this column as the survey weight.
ordinal_threshold	Integer; max unique values for a categorical column to be classified as ordinal (default 10).
binary_threshold	Integer; max unique values for a binary column (default 2).

Value

A named list (the dataset profile) with fields n_rows, n_cols, columns (named list of column profiles), suggested_treatment, suggested_outcome, suggested_weights.

---

morie_dataset_profile_summary_table

Render a human-readable dataset profile summary table.

Description

Plain text only; no rich dependency.

Usage

```
morie_dataset_profile_summary_table(profile)
```

Arguments

profile	A morie_dataset_profile.

Value

Character scalar with embedded newlines.

`morie_dataset_profile_to_list` *Serialize a dataset profile to a plain nested list.*

Description

Suitable for JSON / RDS round-trips.

Usage

```
morie_dataset_profile_to_list(profile)
```

Arguments

`profile`	A `morie_dataset_profile` (output of `morie_dataset_profile()`).

Value

Nested named list.

`morie_dataset_suggest_plan` *Suggest an ordered analysis plan based on a dataset profile.*

Description

Uses the inferred measurement levels, binary indicators, and detected treatment/outcome/weight columns to recommend epidemiological analyses (descriptive profile, propensity scores, IPW-ATE, AIPW, ATT/ATC, double-ML, GATE, survey-weighted estimates).

Usage

```
morie_dataset_suggest_plan(profile)
```

Arguments

`profile`	A `morie_dataset_profile`.

Value

A list of suggestion lists; each has `analysis`, `rationale`, and `required_vars`.

`morie_dataset_summarize_column`
Compute level-appropriate summary statistics for one column.

Description

Interval / ratio columns get mean/sd/min/q25/median/q75/max; nominal / ordinal columns get a `top_counts` list of value -> count for the top ten levels.

Usage

```
morie_dataset_summarize_column(x, level)
```

Arguments

`x`	A vector.
`level`	Inferred measurement level (one of nominal/ordinal/ interval/ratio).

Value

Named list of summary statistics.

`morie_datasets_arcgis_item_by_id`
Generic by-id loader for any ArcGIS Online Feature Service item.

Description

Portal-agnostic sibling to `morie_datasets_tps_arcgis_hub_by_id()`. Works for ANY ArcGIS Online item GUID (not just TPS Hub catalog entries). Same five format paths (json / geojson / csv / shapefile / fgdb).

Usage

```
morie_datasets_arcgis_item_by_id(
  item_id,
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  dest = NULL
)
```

Arguments

`item_id`	32-char hex GUID.
`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	Optional SoQL-style WHERE for the FeatureServer query. Default `"1=1"`.
`max_features`	Optional row cap.
`layer_idx`	Integer layer index (default `0L`).
`dest`	Optional destination path for binary downloads.

Details

The hub_id is ALWAYS resolved live (via the items API) because there's no bundled catalog for non-TPS items. If you find yourself calling this against the same item repeatedly, consider adding a named wrapper (e.g. the shipped `morie_datasets_toronto_zoning_per_neighbourhood` wraps EsriCanadaEducation's `af06159170914808983959df6163fc86` with bundled fixtures for offline use).

Value

A `data.frame` (json / csv), parsed GeoJSON list, or file path (binary).

`morie_datasets_arcgis_item_metadata` *Resolve any ArcGIS Online item id to its FeatureServer URL + canonical metadata.*

Description

Lightweight discovery helper – one network call to the ArcGIS Online items API (`/sharing/rest/content/items/<` returns a single-row data.frame with the same columns the TPS Hub catalog (`morie_datasets_tps_arcgis_hub_la` returns: `hub_id`, `title`, `type`, `feature_server_url`, `owner`, `tags`, `snippet`. Use this when the item is NOT in the bundled TPS catalog (any non-TorontoPoliceService item).

Usage

```
morie_datasets_arcgis_item_metadata(item_id)
```

Arguments

`item_id`	32-char hex GUID for an ArcGIS Online item.

Value

A `data.frame` with one row.

Examples

```
# Vee's Toronto Zoning per Neighbourhood discovery
# m <- morie_datasets_arcgis_item_metadata(
#   "af06159170914808983959df6163fc86")
# m$title  #> "Toronto Zoning per Neighbourhood"
```

---

morie_datasets_arsau_aggregate_summary

Ontario Use-of-Force aggregate summary (5-year 2020-2022, pre-RBDS rollup)

Description

Ontario Use-of-Force aggregate summary (5-year 2020-2022, pre-RBDS rollup)

Usage

```
morie_datasets_arsau_aggregate_summary(offline = TRUE, resource_id = NULL)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN.
`resource_id`	Optional override.

Value

A `data.frame`.

---

morie_datasets_arsau_detailed_dataset

Ontario Use-of-Force detailed dataset (5-year 2020-2022, pre-RBDS)

Description

Ontario Use-of-Force detailed dataset (5-year 2020-2022, pre-RBDS)

Usage

```
morie_datasets_arsau_detailed_dataset(offline = TRUE, resource_id = NULL)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN.
`resource_id`	Optional override.

Value

A `data.frame`.

`morie_datasets_arsau_uof_individual_records`
Ontario Use-of-Force individual records (one row per individual-in-incident)

Description

Ontario Use-of-Force individual records (one row per individual-in-incident)

Usage

```
morie_datasets_arsau_uof_individual_records(
  year = "2024",
  offline = TRUE,
  resource_id = NULL
)
```

Arguments

`year`	Reporting year (`"2023"` or `"2024"`).
`offline`	If `TRUE` (default), read the bundled fixture. If `FALSE`, hit the live CKAN endpoint via httr2.
`resource_id`	Optional CKAN resource id override.

Value

A `data.frame`.

`morie_datasets_arsau_uof_main_records`
Ontario Use-of-Force main records (one row per incident)

Description

Wraps the Ontario Police Use-of-Force Race-Based Data Strategy resource. Offline mode reads a small bundled synthetic fixture from `inst/extdata/arsau_uof_main_records_sample.csv` (5 rows in the canonical 23-column subset of the 65-column upstream schema, clearly stamped `SYNTHETIC-FIXTURE-XXX`). Live mode hits the Ontario CKAN datastore-dump JSON endpoint for the requested reporting year.

Usage

```
morie_datasets_arsau_uof_main_records(
  year = "2024",
  offline = TRUE,
  resource_id = NULL
)
```

Arguments

`year`	Reporting year (`"2023"` or `"2024"`). Honoured only when `offline = FALSE`.
`offline`	If `TRUE` (default), read the bundled fixture. If `FALSE`, hit the live CKAN endpoint via httr2.
`resource_id`	Optional CKAN resource id override.

Value

A `data.frame`.

References

Ontario Open Data Catalogue, "Police Use of Force" (https://data.ontario.ca/dataset/police-use-of-force-race-based-data); Open Government Licence – Ontario.

Examples

```
df <- morie_datasets_arsau_uof_main_records(offline = TRUE)
head(df[, c("IncidentYear", "PoliceService", "IncidentType")])
```

`morie_datasets_arsau_uof_probe_cycle_records`

Ontario Use-of-Force probe-cycle records (one row per CEW cartridge probe per individual-in-incident)

Description

Ontario Use-of-Force probe-cycle records (one row per CEW cartridge probe per individual-in-incident)

Usage

```
morie_datasets_arsau_uof_probe_cycle_records(
  year = "2024",
  offline = TRUE,
  resource_id = NULL
)
```

Arguments

`year`	Reporting year (`"2023"` or `"2024"`).
`offline`	If `TRUE` (default), read the bundled fixture. If `FALSE`, hit the live CKAN endpoint via httr2.
`resource_id`	Optional CKAN resource id override.

Value

A `data.frame`.

`morie_datasets_arsau_uof_weapon_records` *Ontario Use-of-Force weapon records (one row per weapon per individual-in-incident)*

Description

Year 2023 weapon_records is marked INVALID by the Ontario ministry's technical report and has no published CKAN resource; passing `year = "2023"` with `offline = FALSE` raises.

Usage

```
morie_datasets_arsau_uof_weapon_records(
  year = "2024",
  offline = TRUE,
  resource_id = NULL
)
```

Arguments

`year`	Reporting year (`"2023"` or `"2024"`).
`offline`	If `TRUE` (default), read the bundled fixture. If `FALSE`, hit the live CKAN endpoint via httr2.
`resource_id`	Optional CKAN resource id override.

Value

A `data.frame`.

`morie_datasets_bigquery` *Pull a BigQuery table (or filtered slice) as a* `data.frame`.

Description

Requires the `bigrquery` package and Application Default Credentials.

Usage

```
morie_datasets_bigquery(
  project,
  dataset,
  table,
  where = NULL,
  limit = NULL,
  select = "*",
  billing_project = NULL
)
```

Arguments

`project`	Source project (e.g. `"bigquery-public-data"`).
`dataset`	Source dataset (e.g. `"chicago_crime"`).
`table`	Source table (e.g. `"crime"`).
`where`	Raw SQL WHERE clause (without leading `WHERE`).
`limit`	Optional integer `LIMIT`.
`select`	Projection list; defaults to `"*"`.
`billing_project`	GCP project to bill; `NULL` uses ADC-discovered.

Value

A `data.frame`.

`morie_datasets_browse` *Browse + filter the morie cross-portal dataset catalog*

Description

Phase 3DDD4. Convenience wrapper over `morie_dataset_portal_catalog()` that lets callers filter + search by keyword, portal, api_mode, or loader pattern without writing subset expressions by hand.

Usage

```
morie_datasets_browse(
  keyword = NULL,
  portal = NULL,
  api_mode = NULL,
  loader_pattern = NULL,
  keyword_includes_url = FALSE,
  sort_by = c("dataset_key", "source", "n_rows_bundled", "id")
)
```

Arguments

`keyword`	Optional case-insensitive substring to grep against `dataset_key`/`id`/`loader`. `NULL` (default) skips this filter.
`portal`	Optional portal name (see `morie_dataset_portal_catalog()` for the canonical list). Accepts a character vector for multi-portal queries.
`api_mode`	Optional API mode substring to match against the `api_modes` column (e.g., `"soda3"`, `"arcgis"`, `"opendatasoft"`, `"statcan_wds"`, `"manual_download"`). Accepts a character vector.
`loader_pattern`	Optional perl-style regex against the `loader` column (e.g., `"^morie_datasets_tps"`).
`keyword_includes_url`	If `TRUE`, also greps the `dict_url`.
`sort_by`	Sort order: `"dataset_key"` (default), `"source"`, `"n_rows_bundled"` (descending), or `"id"`.

Details

Filters compose with AND semantics. A `keyword` matches against `dataset_key` + `id` + `loader` (case-insensitive). To match anywhere including the dict URL, pass `keyword_includes_url = TRUE`.

Value

A `data.frame` – the filtered subset of the catalog with the same 7-column schema.

Examples

```
# All TPS datasets, alphabetical (offline; reads the cached
# cross-portal catalog -- no network).
tps <- morie_datasets_browse(portal = "tps_arcgis_hub")
nrow(tps)

# Anything mentioning "homicide"
h <- morie_datasets_browse(keyword = "homicide")
head(h$dataset_key)
```

`morie_datasets_calgary_community_crime_stats`
Calgary Community Crime Statistics (sample)

Description

Phase 3FFF3. Bundled 200-row sample of Calgary's per-community per-month crime counts (Socrata id `78gh-n26t`). Covers all 8 canonical CPS categories.

Phase 3FFF3. Bundled 200-row sample of Calgary fire response calls (Socrata id `bdez-pds9`).

Usage

```
morie_datasets_calgary_community_crime_stats(
  offline = TRUE,
  max_features = NULL
)

morie_datasets_calgary_fire_response_calls(offline = TRUE, max_features = NULL)

morie_datasets_calgary_fire_stations(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with `community`, `category`, `crime_count`, `year`, `month`.

`morie_datasets_calgary_open_crime_adjacent_layers`
Calgary Open Data crime-adjacent catalog

Description

Phase 3FFF3. Bundled snapshot of 157 City-of-Calgary Socrata datasets matched on crime-adjacent keywords (crime, police, fire, ambulance, traffic, incident, collision, bylaw, 311).

Usage

```
morie_datasets_calgary_open_crime_adjacent_layers(offline = TRUE)

morie_datasets_edmonton_open_crime_adjacent_layers(offline = TRUE)

morie_datasets_ottawa_open_crime_adjacent_layers(offline = TRUE)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.

Value

A `data.frame` with `soda_id`, `title`, `type`, `search_keyword`.

`morie_datasets_calgary_socrata_by_id`

Fetch a Calgary Open Data Socrata dataset by ID

Description

Phase 3FFF3. Generic SODA2 fetch wrapper for arbitrary Calgary Socrata resources.

Usage

```
morie_datasets_calgary_socrata_by_id(soda_id, limit = 1000L)

morie_datasets_edmonton_socrata_by_id(soda_id, limit = 1000L)
```

Arguments

`soda_id`	4-4 Socrata resource ID.
`limit`	Page size (default 1000).

Value

A `data.frame` of records.

`morie_datasets_chicago_arrests`

City of Chicago Open Data – Arrests feed (`dpt3-jri9`)

Description

Wraps the City of Chicago "Arrests" open dataset (Socrata resource id `dpt3-jri9`; portal landing https://data.cityofchicago.org/Public-Safety/Arrests/dpt3-jri9/about_data). 24 columns covering up to four charges per arrest plus the pipe-concatenated rollup quartet (`charges_statute` / `charges_description` / `charges_type` / `charges_class`).

Usage

```
morie_datasets_chicago_arrests(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`year`	Integer or `NULL`; server-side year filter (uses `date_extract_y(arrest_date) = <year>` SoQL).
`max_features`	Integer or `NULL`; cap on returned rows. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	Logical; if `TRUE` (default, safer post-3EE), read the bundled synthetic frame.
`resource_id`	Optional Socrata resource id override. Accepts the UUID (`dpt3-jri9`, default) or the publisher's alias (`arrests`).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3VV+ dual-mode dispatch.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks. Default `FALSE`.
`page_size`	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Safety net on paginated walks (default 200).
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

Details

Socrata accepts two resource specifiers interchangeably – the numeric/UUID id (`/resource/dpt3-jri9.json`) and the human-readable alias the publisher assigned (`/resource/arrests.json`). morie defaults to the UUID for stability; pass `resource_id = "arrests"` if you want to exercise the alias path.

Offline mode reads a bundled 5-row synthetic fixture (`inst/extdata/chicago_arrests_dpt3_jri9_sample.csv`) carrying the real upstream snake_case schema. Live mode hits the SODA2 endpoint via `.morie_dataset_socrata_fetch()` and honours the 3OO opt-in pagination (`paginate = TRUE`).

Value

A `data.frame` with the documented 24-col Socrata schema.

References

City of Chicago Data Portal, "Arrests" (`dpt3-jri9`).

Examples

```
df <- morie_datasets_chicago_arrests(offline = TRUE)
df$arrest_date
```

`morie_datasets_chicago_community_areas` *Chicago Community Area boundaries (`cauq-8yn6`)*

Description

Wraps the City of Chicago "Boundaries - Community Areas (current)" open dataset (Socrata resource id `cauq-8yn6`; portal landing https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Community-Areas-current-/cauq-8yn6). The 77 canonical Chicago community areas (Rogers Park, West Ridge, Uptown, Lincoln Square, ..., Edgewater). Resolves the `community_area` foreign key carried by every `morie_datasets_chicago_crime()` row.

Usage

```
morie_datasets_chicago_community_areas(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled fixture.
`geometry`	If `TRUE` and `offline = FALSE`, include the `the_geom` MultiPolygon.
`max_features`	Optional row cap.
`resource_id`	Optional view id override (default `"sp34-6z76"`).
`paginate`	Logical; 3OO/3QQ opt-in pagination via `LIMIT n OFFSET m`.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net.
`app_token`	Optional Socrata app token (sent as `X-App-Token`).

Details

SODA3-only (same filtered/derived-view caveat as Wards).

Offline mode reads a bundled 77-row attribute-only fixture (`inst/extdata/chicago_community_areas.csv`: 5 cols – `area_numbe`, `community`, `area_num_1`, `shape_area`, `shape_len`). The `community` column carries the official canonical name in ALL CAPS.

Value

A `data.frame` with 5 attribute cols (offline) or 6 including `the_geom` (live, `geometry = TRUE`).

References

City of Chicago Data Portal, "Boundaries - Community Areas (current)" (`cauq-8yn6`).

Examples

```
df <- morie_datasets_chicago_community_areas(offline = TRUE)
head(df[, c("area_numbe", "community")])
```

`morie_datasets_chicago_crime`

City of Chicago "Crimes – 2001 to Present" feed (`ijzp-q8t2`)

Description

Wraps the City of Chicago "Crimes – 2001 to Present" open dataset (Socrata resource id `ijzp-q8t2`; portal landing https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/about_data). 22-column schema, one row per reported crime incident (except murders, where one row per victim). Data are extracted from the Chicago PD CLEAR system, refreshed daily with a seven-day lag, and addresses are block-only redacted.

Usage

```
morie_datasets_chicago_crime(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`year`	Integer or `NULL`; server-side year filter.
`max_features`	Integer or `NULL`; cap on returned rows. When `paginate = TRUE` this is the total cap across all walked pages.
`offline`	Logical; if `TRUE`, return the bundled synthetic frame.
`mode`	One of `"soda2"` (default) or `"soda3"`. Selects the API path for live mode: • `"soda2"` -> `/resource/<id>.json?$where=...` via `.morie_dataset_socrata_fetch()` (URL-param SoQL grammar). • `"soda3"` -> `/api/v3/views/<id>/query.json?query=SELECT ...` via `.morie_dataset_soda3_query()` (full SoQL passthrough). Both modes return the same 22-column schema; SODA3 is required when a derived/map view is involved (none here, but available for parity with `morie_datasets_chicago_crime_map()`) and for the canonical "single SoQL string" experience.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk pagination in `page_size` chunks. SODA2 uses `$offset`; SODA3 uses `LIMIT page_size OFFSET m` baked into the SoQL.
`page_size`	Integer; per-page row count when `paginate = TRUE`. Default 1,000 (the unauthenticated SODA2 ceiling).
`max_pages`	Integer; safety net on `paginate = TRUE` walks (default 200 -> up to 200,000 rows without an app_token).
`app_token`	Optional Socrata app token (SODA3 only – sent as the `X-App-Token` header; ignored under `mode = "soda2"`).

Details

Scale warning. As of 2026-05 the live feed carries ~8,557,071 rows (8.56M; last refreshed 2026-05-23) – too large for spreadsheet programs and slow even for programmatic pulls without filtering. Always prefer narrowing the query first (`year = ...` server-side filter) or paginating with `paginate = TRUE` + a large `page_size` (and ideally an app_token). A full unfiltered pull at the default `page_size = 1000` would issue ~8,560 requests; with `page_size = 50000` + an app_token it drops to ~172.

Socrata accepts both the numeric id (`/resource/ijzp-q8t2.json`) and the publisher's `crimes` alias (`/resource/crimes.json`). SODA3 endpoints are also available (`/api/v3/views/crimes/query.jso` as are CSV variants (`/resource/crimes.csv`, `/api/v3/views/crimes/query.csv`). morie defaults to SODA2 JSON via the UUID for stability.

Cross-referenced datasets (Chicago Open Data). The 22-col schema carries geographic and crime-classification foreign keys that other Chicago datasets resolve:

beat morie wraps via `morie_datasets_chicago_police_beats()` (`n9it-hstw`).

district morie wraps via `morie_datasets_chicago_police_districts()` (`24zt-jpfn`).

ward morie wraps via `morie_datasets_chicago_wards()` (`sp34-6z76`, 3UU).

community_area morie wraps via `morie_datasets_chicago_community_areas()` (`cauq-8yn6`, 3UU).

iucr / fbi_code morie wraps via `morie_datasets_chicago_iucr_codes()` (`c7ck-438e`, 3UU).

Value

A `data.frame` with the documented Socrata schema.

`morie_datasets_chicago_crime_map` *City of Chicago "Crimes – 2001 to Present – Map" view (`ahwe-kpsy`)*

Description

Wraps the Socrata MAP VIEW derived from the main Crimes feed (parent_fxf = `ijzp-q8t2`). Verified live as `type: map, parent_fxf: [ijzp-q8t2]` via the Socrata catalog API; landing page at https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present-Map/ahwe-kpsy.

Usage

```
morie_datasets_chicago_crime_map(
  date_from = NULL,
  date_to = NULL,
  where = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`date_from`	Lower bound on `date` (inclusive). Accepts a `Date`, `POSIXct`, or ISO-8601 string. `NULL` defaults to one year before `date_to` (matches the upstream Map view's rolling 1-year window).
`date_to`	Upper bound on `date` (exclusive). `NULL` defaults to today.
`where`	Optional additional SoQL `WHERE` fragment ANDed onto the date window. e.g. `"primary_type='HOMICIDE'"`.
`max_features`	Optional total row cap.
`offline`	Logical; if `TRUE` (default), read the bundled 39-col fixture.
`resource_id`	Optional view id override (default `"ahwe-kpsy"`).
`paginate`	Logical; opt-in pagination via baked-in `LIMIT n OFFSET m`.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net on paginated walks.
`app_token`	Optional Socrata app token (sent as `X-App-Token`).

Details

SODA3-only. The SODA2 endpoint `/resource/ahwe-kpsy.json` does technically return HTTP 200 but ships rows as empty objects (`[{}]`) – column resolution doesn't fire on map/filtered views. This loader uses the SODA3 endpoint `/api/v3/views/ahwe-kpsy/query.json?query=SELECT` via `.morie_dataset_soda3_query()`.

The live ahwe-kpsy view returns a **39-column** schema:

- 22 base ijzp-q8t2 columns (id, case_number, date, ..., location)
- 4 reverse-geocoded extras (`location_address`, `location_city`, `location_state`, `location_zip`)
- 4 Socrata-internal metadata cols (`:id`, `:version`, `:created_at`, `:updated_at`)
- 9 `:@computed_region_*` spatial-overlay columns mapping each row to other Chicago boundary layers (wards, community areas, etc.) via Socrata's automatic point-in-polygon computation

Offline mode reads a bundled 5-row 39-col fixture (`inst/extdata/chicago_crime_map_ahwe_kpsy_sample.csv`).

Value

A `data.frame` with the 39-col schema.

References

City of Chicago Data Portal, "Crimes - 2001 to Present - Map" (`ahwe-kpsy`), derived from `ijzp-q8t2`.

Examples

```
df <- morie_datasets_chicago_crime_map(offline = TRUE)
df$primary_type
```

---

morie_datasets_chicago_crime_odata

City of Chicago Crimes feed via OData v4 (`ijzp-q8t2`)

Description

Third Socrata API mode: OData v4 at `/api/odata/v4/<view_id>`, the same protocol Tableau / Power BI / Excel speak natively. Use this when you want morie to consume the Crimes feed the same way those tools do, or when you want server-driven `@odata.nextLink` pagination instead of the client-driven `$offset` walk that SODA2/SODA3 use.

Usage

```
morie_datasets_chicago_crime_odata(
  filter = NULL,
  select = NULL,
  orderby = NULL,
  top = NULL,
  skip = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`filter`	Optional OData `$filter` string (caller-supplied verbatim; see limitation above).
`select`	Optional comma-separated column list.
`orderby`	Optional OData `$orderby`.
`top`	Optional per-request row count (= `$top`).
`skip`	Optional start offset (= `$skip`).
`max_features`	Optional total row cap across pages.
`offline`	Logical; default `TRUE` reads the bundled 22-col `chicago_crime_synthetic.csv` fixture.
`resource_id`	Optional view id override (default `"ijzp-q8t2"`; pass `"crimes"` for the publisher alias).
`paginate`	Logical; if `TRUE`, follow `@odata.nextLink`.
`max_pages`	Safety net on paginated walks.
`app_token`	Optional Socrata app token (sent as `X-App-Token`).

Details

When to reach for which API mode:

Mode	morie wrapper	best for
SODA2	`morie_datasets_chicago_crime()`	base-feed pulls + `$where` filtering
SODA3 (SoQL)	`morie_datasets_chicago_crime_soql()`	arbitrary `SELECT ... WHERE`
SODA3 (map view)	`morie_datasets_chicago_crime_map()`	derived/filtered views (ahwe-kpsy)
OData v4	`morie_datasets_chicago_crime_odata()`	third-party tool ingestion

Known Socrata limitation. `$filter` is unreliable on Socrata's OData implementation – the parser frequently rejects equality filters with `"The types 'Edm.Boolean' and 'Edm.String' (or 'Edm.Decimal') are not compatible."`. `$top` / `$skip` / `$select` / `$orderby` all work; for filtering, use SODA3.

Value

A `data.frame`.

References

Socrata OData docs: https://support.socrata.com/hc/en-us/articles/115005364207-Access-Data-Insigh

Examples

```
df <- morie_datasets_chicago_crime_odata(offline = TRUE)
nrow(df)
```

`morie_datasets_chicago_crime_resolved`
One-call Chicago crime + boundary + dictionary join

Description

Phase 3VV+. Pulls a slice of `morie_datasets_chicago_crime()` and left-joins each of its five canonical foreign keys against the matching resolver dataset shipped in morie:

Usage

```
morie_datasets_chicago_crime_resolved(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL,
  resolvers = c("ward", "community_area", "beat", "district", "iucr")
)
```

Arguments

`year`	Integer or `NULL`; server-side year filter.
`max_features`	Integer or `NULL`; cap on returned rows. When `paginate = TRUE` this is the total cap across all walked pages.
`offline`	Logical; if `TRUE`, return the bundled synthetic frame.
`mode`	One of `"soda2"` (default) or `"soda3"`. Selects the API path for live mode: • `"soda2"` -> `/resource/<id>.json?$where=...` via `.morie_dataset_socrata_fetch()` (URL-param SoQL grammar).

- "soda3" -> /api/v3/views/<id>/query.json?query=SELECT ... via .morie_dataset_soda3_query() (full SoQL passthrough). Both modes return the same 22-column schema; SODA3 is required when a derived/map view is involved (none here, but available for parity with morie_datasets_chicago_crime_map()) and for the canonical "single SoQL string" experience.

paginate — Logical; if TRUE and offline = FALSE, walk pagination in page_size chunks. SODA2 uses $offset; SODA3 uses LIMIT page_size OFFSET m baked into the SoQL.

page_size — Integer; per-page row count when paginate = TRUE. Default 1,000 (the unauthenticated SODA2 ceiling).

max_pages — Integer; safety net on paginate = TRUE walks (default 200 -> up to 200,000 rows without an app_token).

app_token — Optional Socrata app token (SODA3 only – sent as the X-App-Token header; ignored under mode = "soda2").

resolvers — Character subset of the 5 resolver names to join. Default joins all 5. Pass a shorter vector to skip specific joins (e.g. "iucr" only).

Details

crime field	resolver	join key
beat	morie_datasets_chicago_police_beats()	beat == beat_num
district	morie_datasets_chicago_police_districts()	district == dist_num
ward	morie_datasets_chicago_wards()	ward == ward
community_area	morie_datasets_chicago_community_areas()	community_area == area_numbe
iucr	morie_datasets_chicago_iucr_codes()	iucr == iucr

The resolvers are loaded in offline mode (they're all bundled + small), so this analyzer only touches the network for the crime pull itself. Resolver columns are prefixed with the source name (ward_*, community_*, beat_*, district_*, iucr_*) to avoid collisions with the crime schema.

Both mode = "soda2" and mode = "soda3" are honoured for the crime fetch, matching the dual-API design from 3VV+.

Value

A wide data.frame: crime columns first, then the joined resolver columns with their canonical prefixes.

Examples

```
df <- morie_datasets_chicago_crime_resolved(
  offline = TRUE,
  max_features = 5L,
  resolvers = c("ward", "iucr"))
names(df)
```

`morie_datasets_chicago_crime_soql` *City of Chicago Crimes feed – arbitrary-SoQL escape hatch*

Description

Sibling to `morie_datasets_chicago_crime()` but hits the **SODA3** `/api/v3/views/crimes/query.json` endpoint instead of SODA2's `/resource/ijzp-q8t2.json`. The 8.56M-row scale of the base feed makes SODA2's URL-param `$where` clumsy for non-trivial filters; SODA3 lets you send the full SoQL `SELECT ... WHERE ... ORDER BY ...` string in one go.

Usage

```
morie_datasets_chicago_crime_soql(
  where = NULL,
  select = "*",
  order = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`where`	Optional SoQL `WHERE` fragment (without leading `WHERE`). e.g. `"primary_type='HOMICIDE' AND year=2024"`.
`select`	Projection list (default `"*"`).
`order`	Optional SoQL `ORDER BY` fragment.
`max_features`	Optional total row cap.
`offline`	Logical; if `TRUE` (default), read the 22-col `chicago_crime_synthetic.csv` fixture (same one `morie_datasets_chicago_crime()` uses).
`resource_id`	Optional view id override (default `"ijzp-q8t2"`; pass `"crimes"` for the publisher alias path).
`paginate`	Logical; opt-in pagination.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net.
`app_token`	Optional Socrata app token (sent as header).

Value

A `data.frame`.

Examples

```
df <- morie_datasets_chicago_crime_soql(offline = TRUE)
nrow(df)
```

`morie_datasets_chicago_iucr_codes`
Chicago Police Department – Illinois Uniform Crime Reporting (IUCR) code dictionary (`c7ck-438e`)

Description

Wraps the City of Chicago "Chicago Police Department - Illinois Uniform Crime Reporting (IUCR) Codes" reference table (Socrata resource id `c7ck-438e`; portal landing https://data.cityofchicago.org/Public-Safety/Chicago-Police-Department-Illinois-Uniform-Crime-R/c7ck-438e). 410 IUCR codes mapping the `iucr` foreign key carried by every `morie_datasets_chicago_crime()` row to a human-readable description. Five columns:

Usage

```
morie_datasets_chicago_iucr_codes(
  offline = TRUE,
  max_features = NULL,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled full 410-row fixture.
`max_features`	Optional row cap.
`resource_id`	Optional view id override.
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3VV+ dual-mode dispatch.
`paginate`	Logical; 3OO opt-in pagination.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

Details

iucr 4-character IUCR code (e.g. `"110"` for homicide).
primary_description Top-level category (e.g. `"HOMICIDE"`).
secondary_description Subcategory (e.g. `"FIRST DEGREE MURDER"`).
index_code `"I"` (FBI index crime) or other.
active `TRUE` if the code is currently active.

Available via SODA2 (single-shot or paginated) – this is a base dataset, not a filtered view.

Offline mode reads a bundled 410-row complete fixture (`inst/extdata/chicago_iucr_codes.csv`).

Value

A `data.frame`.

References

City of Chicago Data Portal, "Chicago Police Department - Illinois Uniform Crime Reporting (IUCR) Codes" (`c7ck-438e`).

Examples

```
df <- morie_datasets_chicago_iucr_codes(offline = TRUE)
subset(df, primary_description == "HOMICIDE")
```

`morie_datasets_chicago_neighborhoods` *Chicago Neighborhoods boundary (Office of Tourism)*

Description

Wraps the City of Chicago "Boundaries - Neighborhoods" open dataset (Socrata resource id `y6yq-dbs2`; portal landing https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Neighborhoods/bbvz-uum9). 98 neighbourhoods, originally derived from Neighborhoods_2012b (updated 2025-02-20). The City notes these boundaries are approximate and the names are not official.

Usage

```
morie_datasets_chicago_neighborhoods(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

offline	If TRUE (default), read the bundled attribute-only fixture from inst/extdata/chicago_neighbor
geometry	If TRUE and offline = FALSE, include the the_geom MultiPolygon column in the live-mode result.
max_features	Optional cap on returned rows.
resource_id	Optional Socrata resource id override.
mode	One of "soda2" (default JSON resource endpoint) or "soda3" (SoQL query endpoint). 3VV+ dual-mode dispatch.
paginate	Logical; if TRUE and offline = FALSE, walk SODA2 $offset in page_size chunks (3OO).
page_size	Per-page row count when paginating (default 1000, the unauthenticated Socrata ceiling).
max_pages	Safety net on paginated walks (default 200).
app_token	Optional Socrata API app token for higher rate limits; passed as the X-App-Token header.

Details

Offline mode reads a bundled 98-row attribute-only fixture (pri_neigh, sec_neigh, shape_area, shape_len) – the the_geom MultiPolygon column is stripped to keep the bundled size sane (full GeoJSON is ~800 KB). Live mode hits the SODA2 endpoint via .morie_dataset_socrata_fetch() (mockable).

To get the polygons, pass geometry = TRUE in live mode, which includes the SODA2 the_geom column.

Value

A data.frame with 4 attribute columns (offline mode) or 5 cols including the_geom (live mode with geometry = TRUE).

References

City of Chicago Data Portal, "Boundaries - Neighborhoods"; based on Neighborhoods_2012b.

Examples

```
df <- morie_datasets_chicago_neighborhoods(offline = TRUE)
head(df[, c("pri_neigh", "sec_neigh")])
```

`morie_datasets_chicago_police_beats` *Chicago Police Beats (current) boundaries (`n9it-hstw`)*

Description

Wraps the City of Chicago "Boundaries - Police Beats (current)" open dataset (Socrata resource id `n9it-hstw`; portal landing https://data.cityofchicago.org/Public-Safety/Boundaries-Police-Beats-current-/aerh-rz74). Returns 277 Chicago Police beats with their parent sector + district codes (verified live 2026-05). Attribute schema:

Usage

```
morie_datasets_chicago_police_beats(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled fixture.
`geometry`	If `TRUE` and `offline = FALSE`, include `the_geom` (MultiPolygon).
`max_features`	Optional row cap.
`resource_id`	Optional Socrata resource id override (UUID default; pass `"police-beats"` style alias if known).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3VV+ dual-mode dispatch.
`paginate`	Logical; 3OO opt-in pagination.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net on paginated walks.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

Details

beat_num 4-digit beat id (district + 2-digit beat).

beat Within-sector beat sequence number (string).

sector Within-district sector number (string).

district Parent district number (string).

Offline mode reads a bundled attribute-only fixture (`inst/extdata/chicago_police_beats.csv`) – the `the_geom` MultiPolygon column is stripped to keep bundle size sane. Live mode hits the SODA2 JSON endpoint via `.morie_dataset_socrata_fetch()` (mockable); pass `geometry = TRUE` to include `the_geom`. Threads through the 3OO pagination args.

Value

A `data.frame` with 4 attribute cols (offline) or 5 including `the_geom` (live, `geometry = TRUE`).

References

City of Chicago Data Portal, "Boundaries - Police Beats (current)" (`n9it-hstw`).

Examples

```
df <- morie_datasets_chicago_police_beats(offline = TRUE)
head(df)
```

`morie_datasets_chicago_police_districts`
Chicago Police Districts (current) boundaries (`24zt-jpfn`)

Description

Wraps the City of Chicago "Boundaries - Police Districts (current)" open dataset (Socrata resource id `24zt-jpfn`; portal landing https://data.cityofchicago.org/Public-Safety/Boundaries-Police-Districts-current-/fthy-xz3r). Returns 22 active districts (1-12, 14-20, 22, 24, 25) plus the special "31" headquarters polygon. Attribute schema:

Usage

```
morie_datasets_chicago_police_districts(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled fixture.
`geometry`	If `TRUE` and `offline = FALSE`, include `the_geom` (MultiPolygon).
`max_features`	Optional row cap.
`resource_id`	Optional Socrata resource id override.
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3VV+ dual-mode dispatch.
`paginate`	Logical; 3OO opt-in pagination.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net on paginated walks.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

Details

dist_num District number (string, "1"-"31").

dist_label Display label (e.g. `"1ST"`, `"22ND"`).

Offline mode reads a bundled attribute-only fixture (`inst/extdata/chicago_police_districts.csv`). Live mode hits SODA2 JSON; pass `geometry = TRUE` for `the_geom`.

Socrata exposes this dataset in all four format permutations (SODA2 + SODA3, JSON + GeoJSON + CSV):

- SODA2 JSON: `/resource/24zt-jpfn.json`
- SODA2 GeoJSON: `/resource/24zt-jpfn.geojson`
- SODA2 CSV: `/resource/24zt-jpfn.csv`
- SODA3 JSON: `/api/v3/views/24zt-jpfn/query.json`
- SODA3 GeoJSON: `/api/v3/views/24zt-jpfn/query.geojson`

morie defaults to SODA2 JSON; pass an explicit URL via `resource_id` to exercise the others (e.g. for direct sf reads you'd typically hit the GeoJSON variant via `sf::st_read()` yourself rather than going through this loader).

Value

A `data.frame` with 2 attribute cols (offline) or 3 including `the_geom` (live, `geometry = TRUE`).

References

City of Chicago Data Portal, "Boundaries - Police Districts (current)" (`24zt-jpfn`).

Examples

```
df <- morie_datasets_chicago_police_districts(offline = TRUE)
head(df)
```

`morie_datasets_chicago_wards` *Chicago City Council Ward boundaries (`sp34-6z76`)*

Description

Wraps the City of Chicago "Boundaries - Wards (2023-)" open dataset (Socrata resource id `sp34-6z76`; portal landing https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Wards-2023-/sp34-6z76). 50 wards in the current City Council district map. Resolves the `ward` foreign key carried by every `morie_datasets_chicago_crime()` row.

Usage

```
morie_datasets_chicago_wards(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled fixture.
`geometry`	If `TRUE` and `offline = FALSE`, include the `the_geom` MultiPolygon.
`max_features`	Optional row cap.
`resource_id`	Optional view id override (default `"sp34-6z76"`).
`paginate`	Logical; 3OO/3QQ opt-in pagination via `LIMIT n OFFSET m`.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net.
`app_token`	Optional Socrata app token (sent as `X-App-Token`).

Details

SODA3-only. The SODA2 endpoint `/resource/sp34-6z76.json` returns empty objects – this is a filtered/derived view on Socrata. Live mode uses SODA3 (`/api/v3/views/sp34-6z76/query.json`) via `.morie_dataset_soda3_query()`.

Offline mode reads a bundled 50-row attribute-only fixture (`inst/extdata/chicago_wards.csv`: `ward` / `shape_leng` / `shape_area`). Live mode with `geometry = TRUE` also includes the `the_geom` MultiPolygon column.

Value

A `data.frame` with 3 attribute cols (offline) or 4 including `the_geom` (live, `geometry = TRUE`).

References

City of Chicago Data Portal, "Boundaries - Wards (2023-)" (`sp34-6z76`).

Examples

```
df <- morie_datasets_chicago_wards(offline = TRUE)
head(df)
```

`morie_datasets_ckan_package` *Pull every CSV resource of a CKAN package as a list of data frames.*

Description

Pull every CSV resource of a CKAN package as a list of data frames.

Usage

```
morie_datasets_ckan_package(portal, package_id)
```

Arguments

`portal`	Character; CKAN portal base URL.
`package_id`	Character; CKAN package id or slug.

Value

Named list mapping `resource_name -> data.frame`.

`morie_datasets_ckan_search`
Search a CKAN open-data portal by free-text query.

Description

Examples: `"https://open.canada.ca/data"`, `"https://data.ontario.ca"`, `"https://data.gov.uk"`, `"https://data.europa.eu"`.

Usage

```
morie_datasets_ckan_search(portal, query, rows = 50L)
```

Arguments

`portal`	Character; base URL of the CKAN portal.
`query`	Character; free-text search.
`rows`	Integer; max packages to return (default 50).

Value

A `data.frame` of package metadata.

`morie_datasets_corrections_uof_ethnic_origin`
Inmate-participant ethnic origin

Description

Inmate-participant ethnic origin

Usage

```
morie_datasets_corrections_uof_ethnic_origin(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	Logical; `TRUE` (default) reads the bundled real CKAN sample at `inst/extdata/corrections_uof` `FALSE` hits the live CKAN endpoint.
`resource_id`	Optional CKAN resource id override.
`source`	One of `"auto"`, `"live"`, `"bundled"`, `"synthetic"`, `"empty"`; takes precedence over `offline` when supplied.

`morie_datasets_corrections_uof_incident_type`
Incident-type lookup

Description

Incident-type lookup

Usage

```
morie_datasets_corrections_uof_incident_type(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	Logical; `TRUE` (default) reads the bundled real CKAN sample at `inst/extdata/corrections_uof` `FALSE` hits the live CKAN endpoint.
`resource_id`	Optional CKAN resource id override.
`source`	One of `"auto"`, `"live"`, `"bundled"`, `"synthetic"`, `"empty"`; takes precedence over `offline` when supplied.

`morie_datasets_corrections_uof_incidents`
Use-of-force incidents (head dataset)

Description

Use-of-force incidents (head dataset)

Usage

```
morie_datasets_corrections_uof_incidents(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	Logical; `TRUE` (default) reads the bundled real CKAN sample at `inst/extdata/corrections_uof` `FALSE` hits the live CKAN endpoint.
`resource_id`	Optional CKAN resource id override.
`source`	One of `"auto"`, `"live"`, `"bundled"`, `"synthetic"`, `"empty"`; takes precedence over `offline` when supplied.

Value

`data.frame`.

References

https://data.ontario.ca/dataset/use-of-force-in-correctional-institutions; Open Government Licence – Ontario.

---

morie_datasets_corrections_uof_indigenous

Inmate-participant Indigenous identity

Description

Inmate-participant Indigenous identity

Usage

```
morie_datasets_corrections_uof_indigenous(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	Logical; `TRUE` (default) reads the bundled real CKAN sample at `inst/extdata/corrections_uof` `FALSE` hits the live CKAN endpoint.
`resource_id`	Optional CKAN resource id override.
`source`	One of `"auto"`, `"live"`, `"bundled"`, `"synthetic"`, `"empty"`; takes precedence over `offline` when supplied.

---

morie_datasets_corrections_uof_inmate_incident

Inmate-to-incidents bridging table

Description

Inmate-to-incidents bridging table

Usage

```
morie_datasets_corrections_uof_inmate_incident(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_inmate_participant

Inmate-participant demographics (head)

Description

Inmate-participant demographics (head)

Usage

```
morie_datasets_corrections_uof_inmate_participant(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_institution_summary

Institution-level annual incident summary

Description

Institution-level annual incident summary

Usage

```
morie_datasets_corrections_uof_institution_summary(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_location_summary

Location-of-incident annual summary

Description

Location-of-incident annual summary

Usage

```
morie_datasets_corrections_uof_location_summary(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_race

Inmate-participant race

Description

Inmate-participant race

Usage

```
morie_datasets_corrections_uof_race(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_religion

Inmate-participant religion

Description

Inmate-participant religion

Usage

```
morie_datasets_corrections_uof_religion(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_select_incident_summary

Select-incident-type annual summary

Description

Select-incident-type annual summary

Usage

```
morie_datasets_corrections_uof_select_incident_summary(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_corrections_uof_staff_incident *Staff-to-incidents bridging table*

Description

Staff-to-incidents bridging table

Usage

```
morie_datasets_corrections_uof_staff_incident(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	Logical; TRUE (default) reads the bundled real CKAN sample at inst/extdata/corrections_uof FALSE hits the live CKAN endpoint.
resource_id	Optional CKAN resource id override.
source	One of "auto", "live", "bundled", "synthetic", "empty"; takes precedence over offline when supplied.

---

morie_datasets_cpads *Load the Canadian Postsecondary Alcohol and Drug-use Survey (CPADS)*

Description

Resolves a CPADS analysis frame from one of three sources:

Usage

```
morie_datasets_cpads(
  offline = TRUE,
  mode = c("datastore_search", "csv"),
  limit = NULL,
  q = NULL
)
```

Arguments

`offline`	Logical. `TRUE` (default) prefers the bundled fixture for fast/CRAN-safe runs; `FALSE` fetches the live PUMF.
`mode`	Character; one of `"datastore_search"` (default; queryable CKAN datastore endpoint, supports `limit`/`q`) or `"csv"` (whole PUMF CSV download). Ignored when `offline = TRUE`.
`limit`	Integer or `NULL`; row cap, forwarded to `datastore_search` when `mode = "datastore_search"`. `NULL` returns the datastore default (100 rows).
`q`	Character or `NULL`; free-text query forwarded to `datastore_search` (e.g. `q = "title:jones"`). Ignored for `mode = "csv"`.

Details

1. A pre-wrangled local RDS at `morie_cpads_contract()$expected_wrangled_path` (only useful if you've already produced one), OR
2. the bundled 30-row real PUMF sample at `inst/extdata/cpads_pumf_synthetic.csv` (when `offline = TRUE`, the default; carries all 394 raw PUMF columns plus the 11 morie canonical analysis aliases), OR
3. the live PUMF via the CKAN datastore_search API (default, supports `limit`/`q`) or the full PUMF CSV (with `mode = "csv"`): https://open.canada.ca/data/dataset/736fa9b2-62e4-4e31-aea4-51869605b363/resource/d2639429-c304-45a6-90b3-770562f4d46d/download/cpads-2021-2022-pumf2.csv (when `offline = FALSE`).

CPADS is open data published by Health Canada / Statistics Canada (Open Government Licence – Canada). Aggregate dashboards at https://health-infobase.canada.ca/substance-use/reports/cpads/; PUMF user guide: https://open.canada.ca/data/dataset/736fa9b2-62e4-4e31-aea4-51869605b363/resource/a078e4c3-a910-4349-b00e-6ea0d31d391d/download/20212022-cpads-pumf-user-guide.pdf. Sister surveys (CSADS, CSUS, CTADS) are at https://health-infobase.canada.ca/substance-use/.

Value

A `data.frame` carrying every CPADS PUMF column plus morie's canonical analysis aliases (`weight`, `alcohol_past12m`, `heavy_drinking_30d`, `ebac_tot`, `ebac_legal`, `cannabis_any_use`, `age_group`, `gender`, `province_region`, `mental_health`, `physical_health`).

See Also

`morie_cpads_contract()` for the canonical schema + column map; `morie_datasets_load_by_key()` for catalog-wide dispatch.

`morie_datasets_cpd_public_arrests` *Chicago Police Department – Public Arrest Data (2014-2017)*

Description

Wraps the static historical arrests CSV published by the Chicago Police Department at https://www.chicagopolice.org/statistics-data/public-arrest-data/ covering adult and juvenile arrests from 01 JAN 2014 through 31 DEC 2017, with all personally identifying information removed. Ten upper-case-coded columns matching the CPD data dictionary:

Usage

```
morie_datasets_cpd_public_arrests(
  url = NULL,
  offline = TRUE,
  max_features = NULL
)
```

Arguments

`url`	Optional direct-CSV URL. If `NULL` and `offline = FALSE`, the loader errors with a "lookup pending" message pointing at the chicagopolice.org landing page.
`offline`	Logical; if `TRUE` (default), read the bundled synthetic 5-row fixture (`inst/extdata/cpd_public_r`
`max_features`	Integer or `NULL`; cap on returned rows.

Details

ARR_DISTRICT Chicago PD district (geographic boundary).

ARR_BEAT Chicago PD beat (geographic boundary).

ARR_YEAR Calendar year of the arrest.

ARR_MONTH Calendar month of the arrest.

RACE_CODE_CD Perceived race code.

FBI_CODE IUCR/FBI crime category code.

STATUTE ILCS / MCC statute charged.

STAT_DESCR Plain-text statute title.

CHARGE_CLASS_CD ILCS/MCC charge class code.

CHARGE_TYPE_CD "M" = misdemeanour, "F" = felony.

Unlike the SODA2 feeds, CPD publishes this as a single direct CSV download with no documented API; the file URL is not stable across CPD's quarterly republications. morie therefore ships an offline- first loader; pass a `url` for live mode (visit the landing page to find the current direct-CSV URL).

Value

A `data.frame` with the 10-col CPD schema.

References

Chicago Police Department, "Public Arrest Data"; landing page at chicagopolice.org/statistics-data/public- arrest-data/.

Examples

```
df <- morie_datasets_cpd_public_arrests(offline = TRUE)
df$STAT_DESCR
```

`morie_datasets_edmonton_police_stations` *Edmonton Police Station locations*

Description

Phase 3FFF3. Bundled 10-row fixture of Edmonton Police Service station locations (Socrata id `e7aq-scxv`).

Usage

```
morie_datasets_edmonton_police_stations(offline = TRUE, max_features = NULL)

morie_datasets_edmonton_fire_stations(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with `name`, `address`, `latitude`, `longitude`.

`morie_datasets_external_socrata_layers` *List the external Socrata datasets wrapped by morie*

Description

Sibling discovery helper to `morie_datasets_ontario_ckan_layers()`, covering the non-Ontario open-data Socrata portals morie ships offline-mode fixtures + mocked live-mode dispatch for.

Usage

```
morie_datasets_external_socrata_layers()
```

Details

Coverage:

- City of Chicago "Crimes – 2001 to Present" (`ijzp-q8t2`).
- City of Chicago "Arrests" (`dpt3-jri9`; 3PP).
- City of Chicago "Boundaries-Neighborhoods" (`y6yq-dbs2`).
- City of Chicago "Boundaries-Police-Beats (current)" (`n9it-hstw`; 3PP+).
- City of Chicago "Boundaries-Police-Districts (current)" (`24zt-jpfn`; 3PP+).
- City of Chicago "Boundaries-Wards (2023-)" (`sp34-6z76`; 3UU, SODA3-only).
- City of Chicago "Boundaries-Community-Areas (current)" (`cauq-8yn6`; 3UU, SODA3-only).
- City of Chicago "IUCR Code Dictionary" (`c7ck-438e`; 3UU).
- NYC OpenData NYPD Stop, Question and Frisk (SQF) microdata – three published years (2022 = `e4yi-bvqr`, 2023 = `rbed-zzin`, 2024 = `7v9w-k82r`).

All Chicago Socrata endpoints accept both the numeric/UUID specifier (`/resource/<id>.json`) and the publisher's human-readable alias (`/resource/<alias>.json`, e.g. `/resource/arrests.json` or `/resource/crimes.json`). morie's wrappers default to the UUID for stability; pass `resource_id = "<alias>"` to exercise the alias path.

Value

A `data.frame` with columns `dataset_key`, `label`, `portal`, `resource_url`, `fixture`.

`morie_datasets_load_by_key` *Load a dataset by its cross-portal catalog* `dataset_key`

Description

Phase 3EEE4. Single entry point that dispatches to the right loader function for any of the ~550 datasets in `morie_dataset_portal_catalog()`. Lets callers say `morie_datasets_load_by_key("vpd_crime` or `morie_datasets_load_by_key("hub:b4d0...assault")` without remembering whether the relevant loader is `morie_datasets_vpd_crime()` or `morie_datasets_tps_arcgis_hub_by_id()`.

Usage

```
morie_datasets_load_by_key(
  dataset_key,
  offline = TRUE,
  max_features = NULL,
  mode = c("auto", "soda2", "soda3", "odata"),
  app_token = NULL,
  source = NULL
)
```

Arguments

`dataset_key`	A `dataset_key` from `morie_dataset_portal_catalog()`.
`offline`	If `TRUE` (default), prefer bundled fixtures. `FALSE` forces live mode for loaders that support it.
`max_features`	Optional row cap forwarded to the underlying loader.
`mode`	One of `"auto"` (default), `"soda2"`, `"soda3"`, `"odata"`. Only honoured for Socrata-backed sources (chicago / nyc_nypd / nyc_opendata) that have a per-wrapper `mode` argument. `"auto"` lets each underlying loader pick its native default (soda2 today). 3FFF2.
`app_token`	Optional Socrata application token forwarded to SODA3-capable loaders. 3FFF2.
`source`	Optional portal disambiguator (e.g., `"vancouver_opendata"` vs `"toronto_opendata"`) used when the `dataset_key` collides across portals. 3HHH5. Omit when the key is unique (the common case); pass when an ambiguous-key error is raised to pick the intended portal.

Details

Resolution rules:

Bundled-fixture loaders If the catalog's `loader` column names a function that takes no required arguments beyond optionally `offline`/`max_features`, that function is called directly with `offline = offline` + `max_features`.

Generic CKAN dispatchers For Ontario / Montreal / Toronto CKAN catalog entries whose loader is the generic `morie_datasets_*_ckan_resource()`, the `id` (CKAN package_name slug) is resolved to a primary resource via `package_show` + first CSV resource, then fetched.

Generic ArcGIS Hub dispatcher For TPS Hub entries (`source == "tps_arcgis_hub"`), the bare `hub_id` is passed to `morie_datasets_tps_arcgis_hub_by_id()`.

StatCan WDS For statcan_ccjs entries, returns the cube metadata via `morie_datasets_statcan_cube_metada`

Vancouver Opendatasoft For vancouver_opendata entries beyond the 9 bundled fixtures, dispatches to `morie_datasets_vancouver_opendata_by_id()`.

For datasets where the catalog only knows a key + portal (no row-level fixture, no targeted wrapper), `live = FALSE` raises a clear error pointing at the right live-mode dispatcher.

Value

A `data.frame` (or, for StatCan, the WDS metadata list).

Examples

```
# All three calls below resolve to bundled offline fixtures (no
# network). The first call warms the cross-portal catalog cache
# (~2.8s); subsequent calls reuse it (<0.1s each).
df1 <- morie_datasets_load_by_key("vpd_crime")           # 550 rows
df2 <- morie_datasets_load_by_key("nypd_arrests_ytd")    # 5 rows
df3 <- morie_datasets_load_by_key("assault")             # 5 rows
c(vpd = nrow(df1), nypd = nrow(df2), tps_assault = nrow(df3))
```

---

morie_datasets_montreal_ckan_resource

Fetch records from any donnees.montreal.ca CKAN datastore resource

---

Description

Phase 3EEE1. Generic loader that hits CKAN's `datastore_search` endpoint for a given `resource_id`. Useful for any MTL package beyond the bundled SIM sample.

Usage

```
morie_datasets_montreal_ckan_resource(
  resource_id,
  limit = 100L,
  filters = NULL
)
```

Arguments

`resource_id`	CKAN resource UUID (from `package_show`).
`limit`	Page size (CKAN default 100, max varies by host).
`filters`	Optional named list of `{column: value}` filters.

Value

A `data.frame` of records.

Examples

```
## Not run:
# Hypothetical SPVM station boundaries:
df <- morie_datasets_montreal_ckan_resource(
  resource_id = "abc-def-...",
  limit = 50)

## End(Not run)
```

`morie_datasets_montreal_justice_safety_layers`

Donnees Montreal "Loi, justice et securite publique" catalog

Description

Phase 3EEE1. Bundled 23-row snapshot of every CKAN package in the Law / Justice / Public Safety group on donnees.montreal.ca. Includes the SIM fire/EMS interventions dataset, SPVM police station boundaries, municipal regulations, traffic collisions, and ~20 others.

Usage

```
morie_datasets_montreal_justice_safety_layers(offline = TRUE)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV; if `FALSE`, hits `/action/package_search` live.

Value

A `data.frame` with `package_name`, `title`, `num_resources`, `metadata_modified`, `language`, `license`.

Examples

```
cat_df <- morie_datasets_montreal_justice_safety_layers()
nrow(cat_df)  # 23
head(cat_df$title)
```

`morie_datasets_montreal_sim_intervention_types`
SIM intervention TYPE -> French description dictionary

Description

Phase 3EEE1. Bundled lookup table mapping the `INCIDENT_TYPE_DESC` codes used in SIM interventions to their canonical French descriptions (from the dataset's own `type-interventions-descriptions20` sidecar).

Usage

```
morie_datasets_montreal_sim_intervention_types()
```

Value

A `data.frame` with `INCIDENT_TYPE_DESCRIPTION` + `Description`.

Examples

```
d <- morie_datasets_montreal_sim_intervention_types()
nrow(d)
head(d)
```

`morie_datasets_montreal_sim_interventions`
SIM Montreal Fire Service intervention records (sample)

Description

Phase 3EEE1. Bundled stratified 349-row sample (50 rows per DESCRIPTION_GROUPE category) of SIM (Service de securite incendie de Montreal) interventions, drawn from the full 172,899-row open feed for years 2005-2026.

Usage

```
morie_datasets_montreal_sim_interventions(
  offline = TRUE,
  csv_path = NULL,
  max_features = NULL
)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled sample.
`csv_path`	Optional path to a user-downloaded full CSV.
`max_features`	Optional row cap.

Details

Three source modes:

`offline = TRUE` **(default)** Bundled 349-row sample for tests + intro examples.

`csv_path = "..."` Reads a user-downloaded `donneesouvertes-interventions-sim.csv` (or yearly variant) from the CKAN resource link.

Columns (13): INCIDENT_NBR (per-year incident id), CREATION_DATE_TIME, INCIDENT_TYPE_DESC, DESCRIPTION_GROUPE, CASERNE (fire-hall number), NOM_VILLE, NOM_ARROND (arrondissement), DIVISION, NOMBRE_UNITES (vehicles deployed), MTM8_X, MTM8_Y (Quebec MTM zone 8 NAD83 / EPSG:32188), LONGITUDE, LATITUDE (WGS84, obfuscated to intersections per privacy policy).

Value

A `data.frame` with the 13 SIM columns.

References

CKAN package `interventions-service-securite-incendie-montreal`, https://donnees.montreal.ca/dataset/interventions-service-securite-incendie-montreal.

Examples

```
df <- morie_datasets_montreal_sim_interventions(offline = TRUE)
nrow(df)              # 349
table(df$DESCRIPTION_GROUPE)
```

`morie_datasets_namus_missing_persons` *NamUs missing-persons case metadata.*

Description

NamUs missing-persons case metadata.

Usage

```
morie_datasets_namus_missing_persons(
  state = NULL,
  max_features = NULL,
  offline = FALSE
)
```

Arguments

`state`	Character; two-letter US state code or `NULL` (national).
`max_features`	Integer or `NULL`; cap on returned rows.
`offline`	Logical; if `TRUE`, return a bundled synthetic frame.

Value

A `data.frame`.

`morie_datasets_nibrs` *FBI NIBRS offence-event records via the Crime Data Explorer API.*

Description

Requires an API key (`api_key=` or `FBI_CDE_API_KEY` env var).

Usage

```
morie_datasets_nibrs(
  year = NULL,
  max_features = NULL,
  state = NULL,
  offense = NULL,
  api_key = NULL,
  offline = FALSE
)
```

Arguments

`year`	Integer; reporting year (required unless `offline = TRUE`).
`max_features`	Integer or `NULL`; cap on returned rows.
`state`	Character; two-letter US state code, or `NULL` for national.
`offense`	Character; NIBRS offence slug, or `NULL` for all.
`api_key`	Character; FBI CDE API key (or `NULL` -> env var).
`offline`	Logical; if `TRUE`, return a bundled synthetic frame.

Value

A `data.frame`.

`morie_datasets_nist_rds`
NIST Reference Datasets (RDS) catalog metadata.

Description

NIST Reference Datasets (RDS) catalog metadata.

Usage

```
morie_datasets_nist_rds(
  dataset_id = NULL,
  query = NULL,
  max_features = NULL,
  offline = FALSE
)
```

Arguments

`dataset_id`	Character or `NULL`; specific NIST RDS id.
`query`	Character or `NULL`; free-text search.
`max_features`	Integer or `NULL`; cap on returned rows.
`offline`	Logical; if `TRUE`, return a bundled synthetic frame.

Value

A `data.frame` with the NIST RDS catalog schema.

`morie_datasets_nyc_boroughs`
NYC Borough Boundaries (`gthc-hcne`)

Description

Wraps the NYC OpenData "Borough Boundaries" feed (5 NYC boroughs: Manhattan, Bronx, Brooklyn, Queens, Staten Island). Used as the resolver for the `arrest_boro` (1-letter codes) / `boro_nm` (full names) / `patrol_borough_name` foreign keys on NYPD CJ datasets (3NN).

Usage

```
morie_datasets_nyc_boroughs(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

offline	If TRUE (default), read the bundled fixture.
geometry	If TRUE and offline = FALSE, include the the_geom MultiPolygon.
max_features	Optional row cap.
resource_id	Optional view id override (default "sp34-6z76").
mode	One of "soda2" (default JSON resource endpoint) or "soda3" (SoQL query endpoint). 3AAA dual-mode dispatch.
paginate	Logical; 3OO/3QQ opt-in pagination via LIMIT n OFFSET m.
page_size	Per-page row count when paginating.
max_pages	Safety net.
app_token	Optional Socrata app token (sent as X-App-Token).

Details

Attribute schema: borocode (string, "1"-"5"), boroname (capitalised name), shape_area, shape_leng. Live mode also returns the_geom MultiPolygon when geometry = TRUE.

Value

A data.frame.

---

morie_datasets_nyc_boundaries_catalog

Unified catalog of NYC OpenData boundary loaders

Description

Phase 3CCC2. One-stop index of every NYC boundary fixture morie ships, with its loader, SODA id, expected row count, and a note on its join key.

Usage

```
morie_datasets_nyc_boundaries_catalog()
```

Details

NOTE: school/council/community/NTA boundaries are NOT directly row-key joinable to NYPD CJ data – the CJ rows carry lat/long (or just precinct/borough), not a district ID. Use these loaders standalone for geographic context, or pair with a spatial join via the `sf` package on `the_geom` (not bundled to keep morie lightweight).

Value

A `data.frame` with one row per boundary fixture.

Examples

```
morie_datasets_nyc_boundaries_catalog()
```

`morie_datasets_nyc_community_districts`

NYC community district boundaries

Description

Phase 3CCC2. Bundled snapshot of NYC OpenData `5crt-au7u` (71 districts).

Usage

```
morie_datasets_nyc_community_districts(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with `boro_cd`, `shape_leng`, `shape_area`.

`morie_datasets_nyc_council_districts`
NYC City Council district boundaries

Description

Phase 3CCC2. Bundled snapshot of NYC OpenData `872g-cjhh` (51 districts).

Usage

```
morie_datasets_nyc_council_districts(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with `coundist`, `shape_leng`, `shape_area`.

`morie_datasets_nyc_ntas_2020`
NYC Neighborhood Tabulation Areas (2020)

Description

Phase 3CCC2. Bundled snapshot of NYC OpenData `9nt8-h7nd` (262 NTAs from the 2020 census revision). Carries boro + county FIPS + parent CDTA so it can be aggregated upward without spatial intersection.

Usage

```
morie_datasets_nyc_ntas_2020(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with 11 cols including `nta2020`, `ntaname`, `borocode`, `boroname`, `countyfips`, `cdta2020`, `cdtaname`.

`morie_datasets_nyc_nypd_arrests_historic` *NYPD Arrests Data (Historic)*

Description

NYPD Arrests Data (Historic)

Usage

```
morie_datasets_nyc_nypd_arrests_historic(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

`year`	Optional year filter (server-side SoQL).
`max_features`	Optional row cap. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	If `TRUE` (default), read the bundled fixture.
`resource_id`	Optional Socrata resource id override.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks until exhausted or `max_features` is reached. Default `FALSE` (single 1,000-row request, matching the historical pre-3OO behaviour).
`page_size`	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

`morie_datasets_nyc_nypd_arrests_ytd`
NYPD Arrest Data (Year to Date)

Description

NYPD Arrest Data (Year to Date)

Usage

```
morie_datasets_nyc_nypd_arrests_ytd(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

`year`	Optional year filter (server-side SoQL).
`max_features`	Optional row cap. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	If `TRUE` (default), read the bundled fixture.
`resource_id`	Optional Socrata resource id override.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks until exhausted or `max_features` is reached. Default `FALSE` (single 1,000-row request, matching the historical pre-3OO behaviour).
`page_size`	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

`morie_datasets_nyc_nypd_boro_crosswalk`
NYPD borough-code cross-reference (1-letter / UPPER / numeric)

Description

NYPD CJ datasets reference boroughs through three different encodings depending on the table:

Usage

```
morie_datasets_nyc_nypd_boro_crosswalk()
```

Details

`arrest_boro` 1-letter code "B/Q/M/S/K" (Arrests).

`boro_nm` UPPER full name "MANHATTAN" etc. (Complaints).

`borocode / boroname` numeric "1"-"5" + Title-Case (Borough Boundaries gthc-hcne).

This helper returns a 5-row crosswalk between all four forms so callers can left-join NYPD data against the `morie_datasets_nyc_boroughs()` boundary table regardless of which encoding their source uses. Used internally by `morie_datasets_nyc_nypd_resolved()`.

Value

A `data.frame` with 4 columns: `arrest_boro`, `boro_nm`, `borocode`, `boroname`.

`morie_datasets_nyc_nypd_by_key`
Generic NYC NYPD dataset loader by registry key

Description

Generic NYC NYPD dataset loader by registry key

Usage

```
morie_datasets_nyc_nypd_by_key(
  dataset_key,
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
```

```
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

dataset_key	One of the keys in morie_datasets_nyc_nypd_layers().
year	Optional year filter (server-side SoQL).
max_features	Optional row cap. When paginate = TRUE this is the total cap across walked pages.
offline	If TRUE (default), read the bundled fixture.
resource_id	Optional Socrata resource id override.
paginate	Logical; if TRUE and offline = FALSE, walk SODA2 $offset in page_size chunks until exhausted or max_features is reached. Default FALSE (single 1,000-row request, matching the historical pre-3OO behaviour).
page_size	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
max_pages	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
mode	One of "soda2" (default JSON resource endpoint) or "soda3" (SoQL query endpoint). 3AAA dual-mode dispatch.
app_token	Optional Socrata API app token for higher rate limits; passed as the X-App-Token header.

Value

A data.frame.

---

morie_datasets_nyc_nypd_complaint_historic
NYPD Complaint Data Historic

Description

NYPD Complaint Data Historic

Usage

```
morie_datasets_nyc_nypd_complaint_historic(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
```

```
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

`year`	Optional year filter (server-side SoQL).
`max_features`	Optional row cap. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	If `TRUE` (default), read the bundled fixture.
`resource_id`	Optional Socrata resource id override.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks until exhausted or `max_features` is reached. Default `FALSE` (single 1,000-row request, matching the historical pre-3OO behaviour).
`page_size`	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

`morie_datasets_nyc_nypd_complaint_ytd` *NYPD Complaint Data Current (Year To Date)*

Description

NYPD Complaint Data Current (Year To Date)

Usage

```
morie_datasets_nyc_nypd_complaint_ytd(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

`year`	Optional year filter (server-side SoQL).
`max_features`	Optional row cap. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	If `TRUE` (default), read the bundled fixture.
`resource_id`	Optional Socrata resource id override.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks until exhausted or `max_features` is reached. Default `FALSE` (single 1,000-row request, matching the historical pre-3OO behaviour).
`page_size`	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

`morie_datasets_nyc_nypd_hate_crimes` *NYPD Hate Crimes*

Description

NYPD Hate Crimes

Usage

```
morie_datasets_nyc_nypd_hate_crimes(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

year	Optional year filter (server-side SoQL).
max_features	Optional row cap. When paginate = TRUE this is the total cap across walked pages.
offline	If TRUE (default), read the bundled fixture.
resource_id	Optional Socrata resource id override.
paginate	Logical; if TRUE and offline = FALSE, walk SODA2 $offset in page_size chunks until exhausted or max_features is reached. Default FALSE (single 1,000-row request, matching the historical pre-3OO behaviour).
page_size	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
max_pages	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
mode	One of "soda2" (default JSON resource endpoint) or "soda3" (SoQL query endpoint). 3AAA dual-mode dispatch.
app_token	Optional Socrata API app token for higher rate limits; passed as the X-App-Token header.

---

morie_datasets_nyc_nypd_law_books

NYS / NYC statute book code dictionary

Description

Phase 3CCC1. Maps the leading alpha prefix of an NYPD law_code (e.g., "PL" in "PL 1601005") to its human-readable statute book name + jurisdiction (NYS vs NYC). Covers all 22 distinct prefixes observed in the YTD arrests feed + 24 additional canonical NYS / NYC codes that appear in complaint, summons, and historical arrest data.

Usage

```
morie_datasets_nyc_nypd_law_books()
```

Value

A data.frame with columns book, name, jurisdiction.

Examples

```
books <- morie_datasets_nyc_nypd_law_books()
subset(books, book == "PL")
```

`morie_datasets_nyc_nypd_layers`
List the NYPD criminal-justice Socrata datasets wrapped by morie

---

Description

List the NYPD criminal-justice Socrata datasets wrapped by morie

Usage

```
morie_datasets_nyc_nypd_layers()
```

Value

A `data.frame` with 8 columns: `dataset_key`, `label`, `resource_id`, `resource_url`, `permalink` (`data.cityofnewyork.us/d/<id>` stable redirect), `data_dictionary_url` (XLSX, when published as a dataset attachment; `NA_character_` otherwise), `footnotes_url` (PDF, when published; `NA_character_` otherwise), `fixture` (bundled-fixture filename).

Currently only `nypd_arrests_ytd` carries the canonical NYC OpenData attachment URLs (XLSX dictionary + PDF footnotes). The other 7 entries leave those slots `NA`; PRs welcome to fill them in when the asset UUIDs are looked up at the dataset's landing page.

`morie_datasets_nyc_nypd_offense_codes`
NYPD offense-code dictionary (`ky_cd` + `pd_cd` + descriptions)

Description

NYC OpenData does NOT publish a standalone NYPD-offense-code table; the canonical mapping is implicit in the (`ky_cd`, `ofns_desc`, `pd_cd`, `pd_desc`, `law_cat_cd`) tuples carried by every Arrests / Complaints record. This bundled fixture was derived by running a `$group` query on the NYPD Arrests YTD feed (`uip8-fykc`) at fixture-creation time, giving the 246 distinct offense tuples currently in active use. Mirrors the chicago_iucr_codes pattern (3UU).

Usage

```
morie_datasets_nyc_nypd_offense_codes(max_features = NULL)
```

Arguments

`max_features` Optional row cap.

Details

Schema (all character):

ky_cd 3-digit Key Code (top-level offense category).

ofns_desc Description for `ky_cd` (NYPD-truncated to 30 chars; e.g. "MURDER & NON-NEGL. MANSLAUGHTE").

pd_cd 3-digit Penal-Detailed code (subcategory).

pd_desc Description for `pd_cd` (same truncation).

law_cat_cd Penal classification: "F" felony / "M" misdemeanor / "V" violation / "I" infraction / (blank).

The string descriptions ARE truncated at 30 chars in the upstream NYPD feeds; this is NOT a morie processing bug – it's how NYPD's NYS DCJS warehouse stores them. PRs welcome to add a parallel `pd_desc_full` column once a canonical un-truncated source is identified.

Refreshing the fixture:

```
# Re-derive when the YTD feed adds new offense tuples (rare):
#   curl "https://data.cityofnewyork.us/resource/uip8-fykc.json
#         ?$select=ky_cd,ofns_desc,pd_cd,pd_desc,law_cat_cd
#         &$group=ky_cd,ofns_desc,pd_cd,pd_desc,law_cat_cd
#         &$order=ky_cd,pd_cd&$limit=10000"
# then write to inst/extdata/nyc_nypd_offense_codes.csv.
```

Value

A `data.frame` with 246 rows x 5 cols.

Examples

```
codes <- morie_datasets_nyc_nypd_offense_codes()
subset(codes, ky_cd == "104")  # all RAPE subcategories
```

`morie_datasets_nyc_nypd_resolved` *One-call NYPD data + borough + precinct join*

Description

Phase 3AAA. Pulls a slice of any `morie_datasets_nyc_nypd_by_key()`-resolvable dataset and left-joins its borough + precinct foreign keys against the bundled resolvers (`morie_datasets_nyc_boroughs()` + `morie_datasets_nyc_police_precincts()`).

Usage

```
morie_datasets_nyc_nypd_resolved(
  dataset_key,
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL,
  resolvers = c("boro", "precinct", "offense", "law_code")
)
```

Arguments

`dataset_key`	One of the keys in `morie_datasets_nyc_nypd_layers()`.
`year`	Optional year filter (server-side SoQL).
`max_features`	Optional row cap. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	If `TRUE` (default), read the bundled fixture.
`resource_id`	Optional Socrata resource id override.
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks until exhausted or `max_features` is reached. Default `FALSE` (single 1,000-row request, matching the historical pre-3OO behaviour).
`page_size`	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.
`resolvers`	Character subset of `c("boro", "precinct")` to join. Default joins both.

Details

Auto-detects the borough + precinct columns per dataset:

NYPD dataset	boro column	precinct column
nypd_arrests_historic	`arrest_boro` (M/B/K/Q/S)	`arrest_precinct`
nypd_arrests_ytd	`arrest_boro`	`arrest_precinct`
nypd_complaint_historic	`boro_nm` (UPPER)	`addr_pct_cd`
nypd_complaint_ytd	`boro_nm` (UPPER)	`addr_pct_cd`

nypd_hate_crimes	`patrol_borough_name`	`complaint_precinct_code`
nypd_uof_incidents	(none directly; precinct only)	`precinct`

Resolver columns prefixed `boro_*` + `precinct_*` to avoid collisions. Left-join semantics (row count preserved).

Value

A wide `data.frame`: NYPD columns first, then prefixed resolver columns.

Examples

```
df <- morie_datasets_nyc_nypd_resolved("nypd_arrests_ytd",
                                       offline = TRUE)
names(df)
```

---

morie_datasets_nyc_nypd_uof_incidents

NYPD Use of Force Incidents

Description

NYPD Use of Force Incidents

Usage

```
morie_datasets_nyc_nypd_uof_incidents(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

`year`	Optional year filter (server-side SoQL).
`max_features`	Optional row cap. When `paginate = TRUE` this is the total cap across walked pages.
`offline`	If `TRUE` (default), read the bundled fixture.
`resource_id`	Optional Socrata resource id override.

paginate	Logical; if TRUE and offline = FALSE, walk SODA2 $offset in page_size chunks until exhausted or max_features is reached. Default FALSE (single 1,000-row request, matching the historical pre-3OO behaviour).
page_size	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
max_pages	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
mode	One of "soda2" (default JSON resource endpoint) or "soda3" (SoQL query endpoint). 3AAA dual-mode dispatch.
app_token	Optional Socrata API app token for higher rate limits; passed as the X-App-Token header.

`morie_datasets_nyc_nypd_uof_subjects`
NYPD Use of Force: Subjects

Description

NYPD Use of Force: Subjects

Usage

```
morie_datasets_nyc_nypd_uof_subjects(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

year	Optional year filter (server-side SoQL).
max_features	Optional row cap. When paginate = TRUE this is the total cap across walked pages.
offline	If TRUE (default), read the bundled fixture.
resource_id	Optional Socrata resource id override.
paginate	Logical; if TRUE and offline = FALSE, walk SODA2 $offset in page_size chunks until exhausted or max_features is reached. Default FALSE (single 1,000-row request, matching the historical pre-3OO behaviour).

page_size	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
max_pages	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).
mode	One of "soda2" (default JSON resource endpoint) or "soda3" (SoQL query endpoint). 3AAA dual-mode dispatch.
app_token	Optional Socrata API app token for higher rate limits; passed as the X-App-Token header.

`morie_datasets_nyc_nypd_vehicle_stops`
NYPD Vehicle Stop Reports

Description

NYPD Vehicle Stop Reports

Usage

```
morie_datasets_nyc_nypd_vehicle_stops(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  resource_id = NULL,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  mode = c("soda2", "soda3"),
  app_token = NULL
)
```

Arguments

year	Optional year filter (server-side SoQL).
max_features	Optional row cap. When paginate = TRUE this is the total cap across walked pages.
offline	If TRUE (default), read the bundled fixture.
resource_id	Optional Socrata resource id override.
paginate	Logical; if TRUE and offline = FALSE, walk SODA2 $offset in page_size chunks until exhausted or max_features is reached. Default FALSE (single 1,000-row request, matching the historical pre-3OO behaviour).
page_size	Per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
max_pages	Safety net on paginated walks (default 200 -> up to 200,000 rows without an app_token).

`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`app_token`	Optional Socrata API app token for higher rate limits; passed as the `X-App-Token` header.

`morie_datasets_nyc_opendata_bulk_layers`
NYC OpenData bulk catalog (2851 entities)

Description

Phase 3HHH1. Bundled snapshot of every CKAN package on donnees.montreal.ca – substantially broader than the 23-row Loi/Justice/Securite subset from 3EEE1.

Phase 3HHH2. Bundled snapshot of every dataset on opendata.vancouver.ca with richer schema (publisher, theme, license, records_count).

Usage

```
morie_datasets_nyc_opendata_bulk_layers(offline = TRUE)

morie_datasets_chicago_opendata_bulk_layers(offline = TRUE)

morie_datasets_toronto_opendata_bulk_layers(offline = TRUE)

morie_datasets_calgary_opendata_bulk_layers(offline = TRUE)

morie_datasets_edmonton_opendata_bulk_layers(offline = TRUE)

morie_datasets_ottawa_opendata_bulk_layers(offline = TRUE)

morie_datasets_montreal_opendata_bulk_layers(offline = TRUE)

morie_datasets_vancouver_opendata_bulk_layers(offline = TRUE)
```

Arguments

`offline`	If `TRUE` (default), reads bundled CSV.

Value

Tabular catalog snapshot.

`morie_datasets_nyc_police_precincts`
NYC Police Precincts boundary layer (`y76i-bdw7`)

Description

Wraps the NYC OpenData "Police Precincts" feed (77 precincts + the special precinct 22 / Central Park alias = 78 rows in this fixture). Used as the resolver for the `arrest_precinct` / `addr_pct_cd` / `complaint_precinct_code` foreign keys on every NYPD CJ dataset (3NN).

Usage

```
morie_datasets_nyc_police_precincts(
  offline = TRUE,
  geometry = FALSE,
  max_features = NULL,
  resource_id = NULL,
  mode = c("soda2", "soda3"),
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L,
  app_token = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled fixture.
`geometry`	If `TRUE` and `offline = FALSE`, include the `the_geom` MultiPolygon.
`max_features`	Optional row cap.
`resource_id`	Optional view id override (default `"sp34-6z76"`).
`mode`	One of `"soda2"` (default JSON resource endpoint) or `"soda3"` (SoQL `query` endpoint). 3AAA dual-mode dispatch.
`paginate`	Logical; 3OO/3QQ opt-in pagination via `LIMIT n OFFSET m`.
`page_size`	Per-page row count when paginating.
`max_pages`	Safety net.
`app_token`	Optional Socrata app token (sent as `X-App-Token`).

Details

Attribute schema: `precinct` (string, "1"-"123"), `shape_leng`, `shape_area`. Live mode also returns `the_geom` MultiPolygon when `geometry = TRUE`.

Value

A `data.frame`.

`morie_datasets_nyc_school_districts`
NYC public school district boundaries (NYS K-12)

Description

Phase 3CCC2. Bundled snapshot of NYC OpenData `8ugf-3d8u` (33 districts).

Usage

```
morie_datasets_nyc_school_districts(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV; if `FALSE`, fetches via SODA2.
`max_features`	Optional row cap.

Value

A `data.frame` with `schooldist`, `shape_leng`, `shape_area`.

Examples

```
df <- morie_datasets_nyc_school_districts(offline = TRUE)
nrow(df)  # 33
```

`morie_datasets_nyc_socrata_by_id`
Fetch a NYC OpenData Socrata dataset by ID

Description

Fetch a NYC OpenData Socrata dataset by ID

Fetch a Chicago Open Data Socrata dataset by ID

Usage

```
morie_datasets_nyc_socrata_by_id(soda_id, limit = 1000L)

morie_datasets_chicago_socrata_by_id(soda_id, limit = 1000L)
```

Arguments

`soda_id`	4-4 Socrata resource ID.
`limit`	Page size.

Value

A `data.frame` of records.

`morie_datasets_nyc_stop_and_frisk` *NYPD Stop, Question and Frisk (SQF) microdata via NYC OpenData.*

Description

NYPD Stop, Question and Frisk (SQF) microdata via NYC OpenData.

Usage

```
morie_datasets_nyc_stop_and_frisk(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  paginate = FALSE,
  page_size = 1000L,
  max_pages = 200L
)
```

Arguments

`year`	Integer or `NULL`; release year (one of 2022, 2023, 2024). `NULL` defaults to the most-recent registered year.
`max_features`	Integer or `NULL`; cap on returned rows. When `paginate = TRUE` this is the total cap across all walked pages.
`offline`	Logical; if `TRUE`, return the bundled synthetic frame.
`paginate`	Logical; if `TRUE` and `offline = FALSE`, walk SODA2 `$offset` in `page_size` chunks. Default `FALSE`.
`page_size`	Integer; per-page row count when paginating (default 1,000, the unauthenticated SODA2 ceiling).
`max_pages`	Integer; safety net on paginated walks (default 200).

Value

A `data.frame`. Schema is NOT normalised across years.

`morie_datasets_nyc_zctas`
NYC ZIP Code Tabulation Areas (ZCTAs)

Description

Phase 3CCC2. Bundled snapshot of NYC OpenData `35j5-n34v` (221 ZCTAs intersecting NYC). ZCTAs are the Census Bureau's geographic approximation of USPS ZIP code service areas – pair with NYPD address-bearing data via ZIP code lookups for a coarser-than-precinct, finer-than-borough geography.

Usage

```
morie_datasets_nyc_zctas(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with `zcta5`, `arealand`, `areawater`, `centlat`, `centlon`, `intptlat`, `intptlon`.

`morie_datasets_ontario_ckan_by_key`
Generic Ontario CKAN dataset loader (by registry key)

Description

Generic Ontario CKAN dataset loader (by registry key)

Usage

```
morie_datasets_ontario_ckan_by_key(
  dataset_key,
  offline = TRUE,
  resource_id = NULL
)
```

Arguments

`dataset_key`	One of the keys in `morie_datasets_ontario_ckan_layers()` (e.g. `"arsau_uof_main_records_`
`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit the live Ontario CKAN datastore-dump endpoint.
`resource_id`	Optional CKAN resource_id override.

Value

A `data.frame`.

`morie_datasets_ontario_ckan_layers`
List the Ontario CKAN datasets wrapped by morie

Description

Returns the consolidated registry of every Ontario Open Data feed morie ships an offline-mode fixture + mocked live-mode dispatch for. Pair with `morie_datasets_ontario_ckan_by_key()` for generic factory access by `dataset_key`.

Usage

```
morie_datasets_ontario_ckan_layers()
```

Details

Coverage as of this release:

- ARSAU UoF: 2 main + 2 individual + 2 probe_cycle + 1 weapon (2024)
  - 2 5-year aggregates (aggregate_summary + detailed_dataset).
- OTIS: d01 Deaths-in-Custody.
- d02-d07 OTIS deaths variants + OTIS a01/b/c families: known but resource_ids not yet wired in (PRs welcome).

Value

A `data.frame` with columns `dataset_key`, `label`, `resource_id`, `family`, `year`, `fixture`.

`morie_datasets_otis_a01`
Load the OTIS A01 Restrictive-Confinement Detailed Dataset

Description

Thin compatibility shim that delegates to `morie_datasets_otis_a01_restrictive_confinement()`.
The OTIS A01 dataset is published openly at `https://data.ontario.ca/dataset/data-on-inmates-in-ontari`
(Ontario Solicitor General; Open Government Licence – Ontario, CKAN resource id `5a0c5804-a055-4031-9743-73`

Usage

```
morie_datasets_otis_a01(offline = TRUE, ...)
```

Arguments

`offline`	Logical. `TRUE` (default) reads the bundled `otis_a01_restrictive_confinement_sample.csv` fixture. `FALSE` fetches the live CKAN dataset.
`...`	Forwarded to `morie_datasets_otis_a01_restrictive_confinement()`.

Details

Earlier morie versions wrongly claimed this data was FOI-only; that was incorrect and has been retracted as of 3MMM.

Value

A `data.frame`.

See Also

`morie_datasets_otis_a01_restrictive_confinement()`, `morie_datasets_load_by_key()`.

`morie_datasets_otis_a01_restrictive_confinement`

OTIS a01 – Restrictive Confinement (detailed per-individual)

Description

OTIS a01 – Restrictive Confinement (detailed per-individual)

Usage

```
morie_datasets_otis_a01_restrictive_confinement(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

Value

A `data.frame` with the canonical 10-col schema (`EndFiscalYear`, `UniqueIndividual_ID`, `Region_AtTimeOfPlacement`, `Region_MostRecentPlacement`, `Gender`, `Age_Category`, `MentalHealth_Alert`, `SuicideRisk_Alert`, `SuicideWatch_Alert`, `Number_Of_Placements`).

`morie_datasets_otis_b01_segregation_detailed`

OTIS b01 – Segregation detailed (per-individual episodes)

Description

OTIS b01 – Segregation detailed (per-individual episodes)

Usage

```
morie_datasets_otis_b01_segregation_detailed(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

Value

A `data.frame` with the canonical 18-col schema.

`morie_datasets_otis_b02_segregation_total_days`

OTIS b02 – Segregation total days per individual

Description

OTIS b02 – Segregation total days per individual

Usage

```
morie_datasets_otis_b02_segregation_total_days(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	If TRUE (default), read the bundled synthetic fixture. If FALSE, hit Ontario CKAN (resource_id required).
resource_id	Optional CKAN resource id (required for live).
source	One of NULL (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), "real" (force the real CKAN sample bundled in inst/extdata/, error if absent), or "synth" (return a deterministic set.seed() synthetic for didactic examples).

---

morie_datasets_otis_b03_seg_alerts_by_institution

OTIS b03 – Segregation placements: alerts + hold by institution

---

Description

OTIS b03 – Segregation placements: alerts + hold by institution

Usage

```
morie_datasets_otis_b03_seg_alerts_by_institution(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

offline	If TRUE (default), read the bundled synthetic fixture. If FALSE, hit Ontario CKAN (resource_id required).
resource_id	Optional CKAN resource id (required for live).
source	One of NULL (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), "real" (force the real CKAN sample bundled in inst/extdata/, error if absent), or "synth" (return a deterministic set.seed() synthetic for didactic examples).

`morie_datasets_otis_b04_seg_consecutive_by_region`
OTIS b04 – Segregation consecutive durations by region

Description

OTIS b04 – Segregation consecutive durations by region

Usage

```
morie_datasets_otis_b04_seg_consecutive_by_region(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_b05_seg_consecutive_lengths`
OTIS b05 – Segregation placements by consecutive-length bucket

---

Description

OTIS b05 – Segregation placements by consecutive-length bucket

Usage

```
morie_datasets_otis_b05_seg_consecutive_lengths(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_b06_seg_reason_by_institution` *OTIS b06 – Segregation placements: reason for placement by institution*

Description

OTIS b06 – Segregation placements: reason for placement by institution

Usage

```
morie_datasets_otis_b06_seg_reason_by_institution(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_b07_seg_alerts_by_gender`
OTIS b07 – Segregation placements: alerts + hold by gender

Description

OTIS b07 – Segregation placements: alerts + hold by gender

Usage

```
morie_datasets_otis_b07_seg_alerts_by_gender(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_b08_seg_consecutive_by_institution`
OTIS b08 – Segregation consecutive durations by institution

Description

OTIS b08 – Segregation consecutive durations by institution

Usage

```
morie_datasets_otis_b08_seg_consecutive_by_institution(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_b09_seg_n_times`

OTIS b09 – Individuals in segregation by number of times placed

---

Description

OTIS b09 – Individuals in segregation by number of times placed

Usage

```
morie_datasets_otis_b09_seg_n_times(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c01_individuals_total`
OTIS c01 – Total individuals (in custody / restrictive confinement / segregation)

Description

OTIS c01 – Total individuals (in custody / restrictive confinement / segregation)

Usage

```
morie_datasets_otis_c01_individuals_total(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c02_individuals_by_institution`
OTIS c02 – Individuals by institution

Description

OTIS c02 – Individuals by institution

Usage

```
morie_datasets_otis_c02_individuals_by_institution(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c03_individuals_race_by_gender`

OTIS c03 – Individuals by race x gender

Description

OTIS c03 – Individuals by race x gender

Usage

```
morie_datasets_otis_c03_individuals_race_by_gender(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c04_individuals_race_by_region`
OTIS c04 – Individuals by race x region

Description

OTIS c04 – Individuals by race x region

Usage

```
morie_datasets_otis_c04_individuals_race_by_region(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c05_individuals_religion_by_region`
OTIS c05 – Individuals by religion x region

Description

OTIS c05 – Individuals by religion x region

Usage

```
morie_datasets_otis_c05_individuals_religion_by_region(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c06_individuals_age_by_region`

OTIS c06 – Individuals by age category x region

Description

OTIS c06 – Individuals by age category x region

Usage

```
morie_datasets_otis_c06_individuals_age_by_region(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c07_individuals_alerts`
OTIS c07 – Individuals: alerts + hold flags

Description

OTIS c07 – Individuals: alerts + hold flags

Usage

```
morie_datasets_otis_c07_individuals_alerts(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c08_individuals_religion_by_gender`
OTIS c08 – Individuals by religion x gender

Description

OTIS c08 – Individuals by religion x gender

Usage

```
morie_datasets_otis_c08_individuals_religion_by_gender(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c09_individuals_age_by_gender`

OTIS c09 – Individuals by age category x gender

Description

OTIS c09 – Individuals by age category x gender

Usage

```
morie_datasets_otis_c09_individuals_age_by_gender(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c10_aggregate_durations_by_institution`
OTIS c10 – Aggregate durations by institution

Description

OTIS c10 – Aggregate durations by institution

Usage

```
morie_datasets_otis_c10_aggregate_durations_by_institution(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c11_aggregate_lengths`
OTIS c11 – Aggregate lengths

Description

OTIS c11 – Aggregate lengths

Usage

```
morie_datasets_otis_c11_aggregate_lengths(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_c12_aggregate_durations_by_region`

OTIS c12 – Aggregate durations by region

Description

OTIS c12 – Aggregate durations by region

Usage

```
morie_datasets_otis_c12_aggregate_durations_by_region(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_d01_deaths_in_custody` *OTIS Deaths-in-Custody detailed dataset (d01)*

Description

Wraps the Ontario "Data on Inmates in Ontario" d01 resource. Schema: `Year, UniqueIndividual_ID, Region_A`

Usage

```
morie_datasets_otis_d01_deaths_in_custody(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/otis_d01_deaths_in_c` (5 rows). If `FALSE`, hit the live CKAN datastore-dump JSON endpoint for resource id `89e3b63f-5679-4fa4-b98a-fdd2dc486f29`.
`resource_id`	Optional CKAN resource id override.
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

Value

A `data.frame`.

References

Ontario Open Data Catalogue, "Data on Inmates in Ontario" (https://data.ontario.ca/dataset/data-on-inmates-in-ontario); Open Government Licence – Ontario.

Examples

```
df <- morie_datasets_otis_d01_deaths_in_custody(offline = TRUE)
table(df$Region_AtTimeOfDeath)
```

`morie_datasets_otis_d02_deaths_by_gender`
OTIS d02 – Deaths in custody by gender

Description

OTIS d02 – Deaths in custody by gender

Usage

```
morie_datasets_otis_d02_deaths_by_gender(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_d03_deaths_by_race`
OTIS d03 – Deaths in custody by race

Description

OTIS d03 – Deaths in custody by race

Usage

```
morie_datasets_otis_d03_deaths_by_race(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_d04_deaths_by_religion`
OTIS d04 – Deaths in custody by religion

Description

OTIS d04 – Deaths in custody by religion

Usage

```
morie_datasets_otis_d04_deaths_by_religion(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_d05_deaths_by_age_category`
OTIS d05 – Deaths in custody by age category

Description

OTIS d05 – Deaths in custody by age category

Usage

```
morie_datasets_otis_d05_deaths_by_age_category(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_d06_cause_by_alert`
OTIS d06 – Deaths in custody by alert type x institution

Description

OTIS d06 – Deaths in custody by alert type x institution

Usage

```
morie_datasets_otis_d06_cause_by_alert(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_otis_d07_alerts_by_housing_unit`

OTIS d07 – Deaths in custody alerts x housing unit

Description

OTIS d07 – Deaths in custody alerts x housing unit

Usage

```
morie_datasets_otis_d07_alerts_by_housing_unit(
  offline = TRUE,
  resource_id = NULL,
  source = NULL
)
```

Arguments

`offline`	If `TRUE` (default), read the bundled synthetic fixture. If `FALSE`, hit Ontario CKAN (`resource_id` required).
`resource_id`	Optional CKAN resource id (required for live).
`source`	One of `NULL` (default, ships an empty 0-row frame with the documented schema when no real CKAN row is bundled), `"real"` (force the real CKAN sample bundled in `inst/extdata/`, error if absent), or `"synth"` (return a deterministic `set.seed()` synthetic for didactic examples).

`morie_datasets_siu_director_reports`
SIU director's-reports index (legacy PDF anchors).

Description

The SIU re-launched their site in 2025 with a JS-rendered case list; this returns the legacy-pattern anchor frame which may be empty.

Usage

```
morie_datasets_siu_director_reports()
```

Value

A `data.frame` with columns `case_number`, `url`, `posted_date`.

`morie_datasets_siu_report_fields`
Extract structured fields from an SIU director's-report text or URL.

Description

Extract structured fields from an SIU director's-report text or URL.

Usage

```
morie_datasets_siu_report_fields(text_or_url)
```

Arguments

`text_or_url`	Character scalar; either the report text (re-used) or a PDF URL (fetched and parsed first).

Value

A named list with fields `report_id`, `incident_date`, `conclusion`, `sections`.

`morie_datasets_siu_report_text`
Download an SIU director's-report PDF and return its plain text.

Description

Download an SIU director's-report PDF and return its plain text.

Usage

```
morie_datasets_siu_report_text(url = NULL, offline = FALSE)
```

Arguments

`url`	Character; direct PDF URL. Required unless `offline = TRUE`.
`offline`	Logical; if `TRUE`, return the bundled synthetic `24-OFD-001` report text instead of hitting the SIU site.

Value

Character scalar (the plain text).

`morie_datasets_statcan_ccjs_cubes`
Statistics Canada CCJS cube registry (curated subset)

Description

Phase 3DDD3. Bundled 10-row registry of high-traffic Canadian Centre for Justice and Community Safety Statistics cubes published through StatCan CODR – the federal-level complement to morie's provincial loaders (Ontario OTIS, BC VPD, etc.).

Usage

```
morie_datasets_statcan_ccjs_cubes()
```

Value

A `data.frame` with `product_id`, `cube_title_en`, `dimensions`, `frequency`.

Examples

```
cubes <- morie_datasets_statcan_ccjs_cubes()
subset(cubes, grepl("homicide", cube_title_en, ignore.case = TRUE))
```

`morie_datasets_statcan_cube_metadata` *Fetch a StatCan cube's metadata (dimensions + members) via WDS*

Description

Phase 3DDD3. Wraps the `getCubeMetadata` POST endpoint.

Usage

```
morie_datasets_statcan_cube_metadata(product_id, timeout_s = 60L)
```

Arguments

`product_id`	Integer cube ID (e.g., `35100177`).
`timeout_s`	HTTP timeout in seconds.

Value

A list with `status` and `object` (dimensions, members, release info, etc.). Errors if `status != "SUCCESS"`.

Examples

```
## Not run:
meta <- morie_datasets_statcan_cube_metadata(35100177)
meta$object$cubeTitleEn
length(meta$object$dimension)

## End(Not run)
```

`morie_datasets_statcan_full_csv_url` *Get the bulk-CSV download URL for a StatCan cube*

Description

Phase 3DDD3. Wraps the `getFullTableDownloadCSV/<productId>/en` GET endpoint. Returns the temporary download URL for the cube's bulk CSV ZIP. Caller is responsible for downloading the ZIP (typically large – often hundreds of MB).

Usage

```
morie_datasets_statcan_full_csv_url(product_id, language = c("en", "fr"))
```

Arguments

product_id	Integer cube ID.
language	One of "en" or "fr".

Value

Character URL string.

Examples

```
## Not run:
url <- morie_datasets_statcan_full_csv_url(35100177)
# download.file(url, "ccjs_177.zip")

## End(Not run)
```

---

morie_datasets_statcan_vectors

Fetch the latest N periods for a set of StatCan vectors via WDS

Description

Phase 3DDD3. Wraps the getDataFromVectorsAndLatestNPeriods POST endpoint. A "vector" is StatCan's atomic time series ID (e.g., v109502878 for "Canada – Total, all violations").

Usage

```
morie_datasets_statcan_vectors(vector_ids, n_periods = 5L, timeout_s = 60L)
```

Arguments

vector_ids	Integer vector of StatCan vector IDs (no v prefix; the API takes raw integers).
n_periods	Number of latest periods per vector (default 5).
timeout_s	HTTP timeout.

Value

A data.frame with one row per (vector, period) observation: vector_id, coordinate, ref_period, value, decimals, status, symbol, scalar_factor.

Examples

```
## Not run:
df <- morie_datasets_statcan_vectors(c(109502878L, 109502879L),
                                     n_periods = 3)
nrow(df)  # ~6

## End(Not run)
```

`morie_datasets_summary` *Tally the cross-portal catalog by source + api_mode*

Description

Phase 3DDD4. Returns a compact summary of how many datasets each portal contributes + which API protocols are used. Useful as a quick "what does morie ship?" smoke test.

Usage

```
morie_datasets_summary()
```

Value

A `data.frame` with one row per portal: `source`, `n_datasets`, `api_modes`, `n_with_bundled_fixture`.

Examples

```
morie_datasets_summary()
```

`morie_datasets_toronto_ambulance_stations` *Toronto Ambulance station locations*

Description

Phase 3EEE2. Bundled snapshot of `ambulance-station-locations` (46 EMS stations across Toronto). Useful as a control overlay for crime + EMS dispatch analyses.

Usage

```
morie_datasets_toronto_ambulance_stations(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with full station address + EMS metadata.

`morie_datasets_toronto_asr_miscellaneous`
TPS Annual Statistical Report – Miscellaneous data (aggregated)

Description

Phase 3EEE2. Bundled snapshot of `police-annual-statistical-report-miscellaneous-data` – 40 rows of year x section x category x subtype aggregates covering hate crime counts, IMPACT calls, and other Toronto Police aggregates that aren't in the per-incident ArcGIS Hub layers.

Usage

```
morie_datasets_toronto_asr_miscellaneous(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads bundled CSV.
`max_features`	Optional row cap.

Value

A `data.frame` with `YEAR`, `SECTION`, `CATEGORY`, `SUBTYPE`, `COUNT_`.

`morie_datasets_toronto_open_ckan_resource`
Fetch records from any open.toronto.ca CKAN resource

Description

Phase 3EEE2. Generic loader hitting CKAN's `datastore_search` endpoint for an arbitrary Toronto package resource.

Usage

```
morie_datasets_toronto_open_ckan_resource(resource_id, limit = 100L)
```

Arguments

`resource_id`	CKAN resource UUID (from `package_show`).
`limit`	Page size (max 32000 per CKAN; sane default 100).

Value

A `data.frame` of records.

morie_datasets_toronto_open_crime_adjacent_layers

Toronto Open Data crime-adjacent CKAN catalog

Description

Phase 3EEE2. Bundled snapshot of 208 City-of-Toronto CKAN packages matched on crime-adjacent keywords (311, fire, police, ambulance, parking, traffic collision, by-law, emergency, crime, wellbeing). Each row identifies a package by its CKAN slug; callers fetch records via `morie_datasets_toronto_open_ckan_resource()` or visit the open.toronto.ca dataset page.

Usage

```
morie_datasets_toronto_open_crime_adjacent_layers(offline = TRUE)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.

Value

A `data.frame` with `package_name`, `title`, `num_resources`, `metadata_modified`, `search_keyword`.

morie_datasets_toronto_zoning_per_neighbourhood

Toronto Zoning per Neighbourhood (EsriCanadaEducation)

Description

Wraps EsriCanadaEducation's ArcGIS Online Feature Service `ZonesofToronto_Neighbourhoods` (item id `af06159170914808983959df6163fc86`; FeatureServer at `services.arcgis.com/As5CFN3ThbQpy8Ph/.../`
Two layers in the service:

Usage

```
morie_datasets_toronto_zoning_per_neighbourhood(
  layer = c("neighbourhoods", "zoning_stats"),
  format = "json",
  where = "1=1",
  max_features = NULL,
  offline = TRUE
)
```

Arguments

`layer`	One of `"neighbourhoods"` (default, polygon demographics) or `"zoning_stats"` (per-zone area table).
`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`. Only honoured when `offline = FALSE`.
`where`	Optional FeatureServer WHERE filter (live mode).
`max_features`	Optional row cap.
`offline`	Logical; if `TRUE` (default), read the bundled synthetic fixture.

Details

`layer = "neighbourhoods"` **(FeatureServer layer 0)** Polygon boundaries for Toronto neighbourhoods with a 39-column demographic schema – total population, sex split, 18 age brackets (0-4 through 85+), senior + youth + child aggregates, and 10 specific language counts (Chinese, Italian, Korean, Persian, Portuguese, Russian, Spanish, Tagalog, Tamil, Urdu) plus a `HomeLanguageCategory` total.

`layer = "zoning_stats"` **(FeatureServer table 1)** Per- neighbourhood zoning-area stats – 4 columns (`OBJECTID`, `ZoneDesc`, `Neighbourhood_Name`, `SUM_Area`). Many rows per neighbourhood, one per `ZoneDesc` (Commercial, Residential, Industrial, etc.).

Offline mode reads bundled 5-row synthetic fixtures (`toronto_zoning_neighbourhoods_sample.csv` / `toronto_zoning_stats_sample.csv`) – SYNTH-stamped, not attributable to actual Toronto neighbourhoods. Live mode hits the FeatureServer via the 3SS+ generic `morie_datasets_arcgis_item_by` resolver.

Value

A `data.frame` (json / csv / offline), parsed GeoJSON list, or file path (binary).

References

Esri Canada Education – ArcGIS Online item `af06159170914808983959df6163fc86`.

Examples

```
df <- morie_datasets_toronto_zoning_per_neighbourhood(offline = TRUE)
head(df[, c("Neighbourhood", "Total_Population", "Seniors65andover")])
```

`morie_datasets_tps_2008_firs`

2008 FIRS

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `b8e3ef826ea84cbcb85951d051afc2fa`.

Usage

```
morie_datasets_tps_2008_firs(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

2008 Field Information Requests (FIRS)

Tags: FIRS; Field Information Requests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_2009_firs` *2009 FIRS*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `de8ba3b4899b48bc8fbf4421f4945ed6`.

Usage

```
morie_datasets_tps_2009_firs(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

2009 Field Information Requests (FIRS)

Tags: FIRS; Freedom of Information Requests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_2010_firs` *2010 FIRS*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `9a5c8a4fdfa54e7f97236a769b196f16`.

Usage

```
morie_datasets_tps_2010_firs(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

2010 Field Information Requests (FIRS)

Tags: FIRS; Field Information Requests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_2011_firs`

2011 FIRS

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `78361ed81cca40aebd1032a26ef52e5b`.

Usage

```
morie_datasets_tps_2011_firs(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

2011 Field Information Requests (FIRS)

Tags: FIRS; Field Information Requests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_2012_firs`

2012 FIRS

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `7a690ba1d7714063983ed78024d5b2af`.

Usage

```
morie_datasets_tps_2012_firs(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

2012 Field Information Requests (FIRS)

Tags: FIRS; Freedom of Information Requests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_2013_firs` *2013 FIRS*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `95a29d4453894944be7a79f537a432b1`.

Usage

```
morie_datasets_tps_2013_firs(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

2013 Field Information Requests (FIRS)

Tags: FIRS; Field Information Requests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_administrative`

Administrative (ASR-AD-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `135607d9250b4e5ea930e7ea39780a77`.

Usage

```
morie_datasets_tps_administrative(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a breakdown of administrative information. This data is compiled and provided by several units of the Toronto Police Service.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Administrative

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_arcgis_hub_by_id`

Generic TPS ArcGIS Hub dataset loader by hub_id

Description

Single entry point for the 71 datasets listed by `morie_datasets_tps_arcgis_hub_layers()`. Supports five export formats:

Usage

```
morie_datasets_tps_arcgis_hub_by_id(
  hub_id,
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`hub_id`	32-char hex GUID. See `morie_datasets_tps_arcgis_hub_layers()` for the canonical 71.
`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

`"json"` **(default)** Hits the FeatureServer `/0/query?where=...&outFields=*&f=json` endpoint and parses attributes into a tidy `data.frame`. Same path the existing TPS PSDP loaders (3FF) use; honours an arbitrary `where` clause and `max_features` cap.

`"geojson"` Hits `?f=geojson` and returns the raw GeoJSON as a parsed R list. Caller can pass to `sf::st_read()`.

`"csv"` Hits the Hub CSV exporter at `hub.arcgis.com/api/v3/datasets/<hub_id>_0/downloads/data?format=` and parses the returned CSV into a `data.frame`.

`"shapefile"` / `"fgdb"` Downloads the binary archive (Esri Shapefile zip / File Geodatabase zip) to `dest` (default `tempfile()`) and returns the file path. Caller can `sf::st_read()` the unzipped contents.

For boundary layers (Police Divisions, Patrol Zone, Facilities) you'll typically want `format = "geojson"` to get the polygon geometry. For tabular outputs (UoF tables, KSI counts, ASR tables, budgets) `format = "json"` is sufficient and lightest.

Value

A `data.frame` (json / csv), a parsed GeoJSON list, or a file path (binary).

morie_datasets_tps_arcgis_hub_download

Direct multi-format downloader (binary or text) for a TPS Hub item

Description

Thin wrapper that just hits the Hub downloads endpoint without the FeatureServer `/query` round-trip. Use this when you want the canonical CSV / GeoJSON / Shapefile / FGDB exactly as the Hub UI serves them (including any column-name and projection differences that on-the-fly exports introduce vs the FeatureServer source).

Usage

```
morie_datasets_tps_arcgis_hub_download(
  hub_id,
  format = "csv",
  layer_idx = 0L,
  dest = NULL
)
```

Arguments

`hub_id`	32-char hex GUID.
`format`	One of `"csv"`, `"geojson"`, `"shapefile"`, `"fgdb"`.
`layer_idx`	Integer layer index (default `0L`).
`dest`	Optional destination path; defaults to `tempfile()`.

Value

Path to the downloaded file.

morie_datasets_tps_arcgis_hub_layers

List the Toronto Police Service ArcGIS Hub datasets wrapped by morie

Description

Sibling discovery helper to `morie_datasets_external_socrata_layers()` and `morie_datasets_ontario_ckan_l` covering all 71 datasets currently published on `data.tps.ca`.

Usage

```
morie_datasets_tps_arcgis_hub_layers(offline = TRUE)
```

Arguments

`offline`	If `TRUE` (default), read the bundled 71-row catalog fixture (`inst/extdata/tps_arcgis_hub_cata` Live mode hits the TPS Hub search API.

Details

Verified live against the canonical TPS Hub catalog API (`https://data.tps.ca/api/search/v1/collections/da` on 2026-05-24 – returned `numberMatched: 71`, all owned by `TorontoPoliceService`.

Coverage spans nine families:

MCI Open Data Assault, Auto Theft, Bicycle Thefts, Break and Enter, Hate Crimes, Homicides, Intimate Partner + Family Violence, Robbery, Shooting + Firearm Discharges, Theft From Motor Vehicle, Theft Over.

Use of Force (RBDC-UOF) Six per-axis breakdown tables plus the Types-and-Perceived-Weapons crosstab.

Annual Statistical Report (ASR) Administrative, Calls for Service, Enforcement, Firearms, Misc., Personnel, Public Complaints, Reported Crimes, Regulated Interactions, Search of Persons, Traffic, Victims of Crime.

Killed and Seriously Injured (KSI) Main + per-mode (Automobile / Cyclist / Fatals / Motorcyclist / Passenger / Pedestrian).

Budget Annual figures 2020 through 2026 plus `Budget_by_Command`.

Personnel + Staffing ASR-PB family + `Staffing_by_Command`.

Historical FIRS Annual files 2008 through 2013.

Geographic boundaries Facilities, Patrol Zone, Police Divisions.

Other Community Safety Indicators, Mental Health Act Apprehensions, Neighbourhood Crime Rates, Persons in Crisis Calls for Service Attended.

Value

A `data.frame` with columns `hub_id`, `title`, `type`, `feature_server_url`, `owner`, `tags`, `snippet`.

References

TPS Public Safety Data Portal, https://data.tps.ca/search?collection=dataset.

Examples

```
cat <- morie_datasets_tps_arcgis_hub_layers()
nrow(cat)        # 71
head(cat$title)
```

`morie_datasets_tps_arrested_and_charged_persons`
Arrested and Charged Persons (ASR-ENF-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `4702e79fd2404f7d93dd9866f45d7ec2`.

Usage

```
morie_datasets_tps_arrested_and_charged_persons(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides an aggregate count of persons who have been arrested and charged. The data is aggregated by division, neighbourhood, sex, age, crime category, and crime subtype.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Arrests

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_arrests_and_strip_searches`
Arrests and Strip Searches (RBDC-ARR-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `899f1b3b047c47659a9843e9c5858269`.

Usage

```
morie_datasets_tps_arrests_and_strip_searches(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset includes information related to all arrests and strip searches.

Tags: RBDC; race; race based data

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_assault`
TPS PSDP – Assault

Description

TPS PSDP – Assault

Usage

```
morie_datasets_tps_assault(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

`year`	Optional reporting year filter (applies an `OCC_YEAR = <year>` WHERE clause when `offline = FALSE`).
`max_features`	Optional cap on returned rows (`offline = FALSE` only).
`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/tps_mha_apprehension` (5 rows in the canonical TPS PSDP 22-column schema). If `FALSE`, hit the TPS PSDP ArcGIS FeatureServer.
`layer_url`	Optional ArcGIS layer URL override.

`morie_datasets_tps_automobile_ksi`
Automobile KSI

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `9a21cd6f550748c3a25ac89a108ca5c5`.

Usage

```
morie_datasets_tps_automobile_ksi(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Automobile-related KSI collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; Automobile; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_autotheft`

TPS PSDP – Auto Theft

Description

TPS PSDP – Auto Theft

Usage

```
morie_datasets_tps_autotheft(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

year	Optional reporting year filter (applies an OCC_YEAR = <year> WHERE clause when offline = FALSE).
max_features	Optional cap on returned rows (offline = FALSE only).
offline	If TRUE (default), read the bundled synthetic fixture from inst/extdata/tps_mha_apprehension (5 rows in the canonical TPS PSDP 22-column schema). If FALSE, hit the TPS PSDP ArcGIS FeatureServer.
layer_url	Optional ArcGIS layer URL override.

---

morie_datasets_tps_bicycle_thefts

Bicycle Thefts Open Data

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to morie_datasets_tps_arcgis_hub_by_id() with the canonical hub item_id a89d10d5e28444ceb0c8d1d4c0ee39cc.

Usage

```
morie_datasets_tps_bicycle_thefts(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

format	One of "json" (default), "geojson", "csv", "shapefile", "fgdb".
where	SoQL-style WHERE filter passed to FeatureServer ?where= (default "1=1").
max_features	Optional cap on returned rows. Passed as resultRecordCount to FeatureServer; ignored for binary formats.
layer_idx	Integer index of the FeatureServer layer to pull (default 0L, the first layer).
offline	Logical; if TRUE, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default TRUE – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
dest	Optional path for binary downloads (format %in% c("shapefile", "fgdb")); defaults to tempfile().

Details

Bicycle Theft occurrences by reported date.

Tags: Bike; Bicycle; Thefts; Crime; Toronto; TPS; Toronto Police

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_bicycletheft`
TPS PSDP – Bicycle Theft

Description

TPS PSDP – Bicycle Theft

Usage

```
morie_datasets_tps_bicycletheft(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

`year`	Optional reporting year filter (applies an `OCC_YEAR = <year>` WHERE clause when `offline = FALSE`).
`max_features`	Optional cap on returned rows (`offline = FALSE` only).
`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/tps_mha_apprehension` (5 rows in the canonical TPS PSDP 22-column schema). If `FALSE`, hit the TPS PSDP ArcGIS FeatureServer.
`layer_url`	Optional ArcGIS layer URL override.

`morie_datasets_tps_breakandenter`
TPS PSDP – Break and Enter

Description

TPS PSDP – Break and Enter

Usage

```
morie_datasets_tps_breakandenter(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

`year`	Optional reporting year filter (applies an `OCC_YEAR = <year>` WHERE clause when `offline = FALSE`).
`max_features`	Optional cap on returned rows (`offline = FALSE` only).
`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/tps_mha_apprehension` (5 rows in the canonical TPS PSDP 22-column schema). If `FALSE`, hit the TPS PSDP ArcGIS FeatureServer.
`layer_url`	Optional ArcGIS layer URL override.

`morie_datasets_tps_budget_2020`
Budget_2020

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `daca9df799ea4a54a29955ce7fb972d4`.

Usage

```
morie_datasets_tps_budget_2020(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_2021`

Budget_2021

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `b511c476865b4f0a993cb7bb1c6be7cf`.

Usage

```
morie_datasets_tps_budget_2021(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_2022`

Budget_2022

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `25c20a7f15e44579acb947510405ab24`.

Usage

```
morie_datasets_tps_budget_2022(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_2023`

Budget_2023

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `6ac1f56513ab481091ce16f435c390b7`.

Usage

```
morie_datasets_tps_budget_2023(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_2024`

Budget_2024

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `584b12967d214bb9a673505d97295eea`.

Usage

```
morie_datasets_tps_budget_2024(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_2025`

Budget_2025

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `cae4c92e80f84e949de156b3ac0d4fef`.

Usage

```
morie_datasets_tps_budget_2025(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_2026`

Budget_2026

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `d80f9e0b3cc74f649e5e4593cdda207e`.

Usage

```
morie_datasets_tps_budget_2026(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_budget_by_command` *Budget_by_Command*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `3dca9164b32e4ca7b9c23f41efc9904b`.

Usage

```
morie_datasets_tps_budget_by_command(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_calls_for_service_attended`

Calls for Service Attended (ASR-CS-TBL-003)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `46c7581a136445c78831acb657a4fb0d`.

Usage

```
morie_datasets_tps_calls_for_service_attended(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a count of calls for service attended aggregated by division and neighbourhood.

Tags: ASR; TPS; Annual Statistical Report; Toronto Police; Calls for Service

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_community_safety_indicators`

Community Safety Indicators Open Data

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `0a239a5563a344a3bbf8452504ed8d68`.

Usage

```
morie_datasets_tps_community_safety_indicators(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Community Safety Indicators (CSI) occurrences by reported date.

Tags: Community Safety Indicators; CSI; Crime; Toronto; TPS; Toronto Police

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_complaint_dispositions`

Complaint Dispositions (ASR-PCF-TBL-003)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `8f3cbe34f3724f93b3aa321b3e957092`.

Usage

```
morie_datasets_tps_complaint_dispositions(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a breakdown of the total investigated complaints by disposition of the complaint submitted.

Tags: ASR; TPS; Toronto Police; Open Data; Annual Statistical Report; Complaints

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_cyclist_ksi` *Cyclist KSI*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `b38c2524696943bb86398d314a06a42a`.

Usage

```
morie_datasets_tps_cyclist_ksi(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

format	One of "json" (default), "geojson", "csv", "shapefile", "fgdb".
where	SoQL-style WHERE filter passed to FeatureServer ?where= (default "1=1").
max_features	Optional cap on returned rows. Passed as resultRecordCount to FeatureServer; ignored for binary formats.
layer_idx	Integer index of the FeatureServer layer to pull (default 0L, the first layer).
offline	Logical; if TRUE, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default TRUE – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
dest	Optional path for binary downloads (format %in% c("shapefile", "fgdb")); defaults to tempfile().

Details

Cyclist-related KSI collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; Cyclist; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see morie_datasets_tps_arcgis_hub_by_id() for per-format semantics.

---

morie_datasets_tps_dispatched_calls_by_division

Dispatched Calls by Division (ASR-CS-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to morie_datasets_tps_arcgis_hub_by_id() with the canonical hub item_id 9cfcd6fe0d374f65afda69c4b9bdc60a.

Usage

```
morie_datasets_tps_dispatched_calls_by_division(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a count of the dispatched calls by division, including some specific units such as PRIME, Parking and "Other". This data includes the command level at the time of reporting.

Tags: ASR; TPS; Annual Statistical Report; Toronto Police; Dispatched Calls

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_facilities`

Facilities

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `6288c8314c594bc9a384df2cf17f8474`.

Usage

```
morie_datasets_tps_facilities(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Police stations and other TPS facilities.

Tags: Police Facilities; TPS Facilities; Facilities

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_fatals_ksi` *Fatals KSI*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `317e768682d14fad94de83eaefbf5954`.

Usage

```
morie_datasets_tps_fatals_ksi(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Fatal-related KSI collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; Fatal; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_firearms_top_calibres`

Firearms Top Calibres (ASR-F-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `9b1b38ed56764b968c25ce6b74e5dc0d`.

Usage

```
morie_datasets_tps_firearms_top_calibres(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

The dataset provides a list of the most common types of pistols, revolvers, rifles and shotguns that comprise the crime guns seized by the Toronto Police Service.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Firearms

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_gross_expenditures_by_division`

Gross Expenditures by Division (ASR-PB-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `6cb7e76c7d5b4bf5bce0c533ca7fdf40`.

Usage

```
morie_datasets_tps_gross_expenditures_by_division(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a breakdown of the Gross Expenditures for each division. This data includes the command level at the time of reporting.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Budget

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_gross_operating_budget`

Gross Operating Budget (ASR-PB-TBL-005)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `8e95b932cd2d404b9d9d26c2ecc8ebec`.

Usage

```
morie_datasets_tps_gross_operating_budget(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides the Gross Operating Budget incurred by the Toronto Police Service.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Budget

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_hatecrimes`

TPS PSDP – Hate Crimes

Description

TPS PSDP – Hate Crimes

Usage

```
morie_datasets_tps_hatecrimes(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

year	Optional reporting year filter (applies an OCC_YEAR = <year> WHERE clause when offline = FALSE).
max_features	Optional cap on returned rows (offline = FALSE only).
offline	If TRUE (default), read the bundled synthetic fixture from inst/extdata/tps_mha_apprehension (5 rows in the canonical TPS PSDP 22-column schema). If FALSE, hit the TPS PSDP ArcGIS FeatureServer.
layer_url	Optional ArcGIS layer URL override.

---

morie_datasets_tps_homicide

TPS Homicides feed.

Description

TPS Homicides feed.

Usage

```
morie_datasets_tps_homicide(year = NULL, max_features = NULL)
```

Arguments

year	Integer or NULL. If set, filter to OCC_YEAR == year server-side.
max_features	Integer or NULL; cap on returned rows.

Value

A data.frame.

---

morie_datasets_tps_homicides

TPS PSDP – Homicides

Description

TPS PSDP – Homicides

Usage

```
morie_datasets_tps_homicides(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

year	Optional reporting year filter (applies an OCC_YEAR = <year> WHERE clause when offline = FALSE).
max_features	Optional cap on returned rows (offline = FALSE only).
offline	If TRUE (default), read the bundled synthetic fixture from inst/extdata/tps_mha_apprehension (5 rows in the canonical TPS PSDP 22-column schema). If FALSE, hit the TPS PSDP ArcGIS FeatureServer.
layer_url	Optional ArcGIS layer URL override.

---

morie_datasets_tps_intimate_partner_family_violence

TPS PSDP – Intimate Partner and Family Violence

Description

TPS PSDP – Intimate Partner and Family Violence

Usage

```
morie_datasets_tps_intimate_partner_family_violence(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

year	Optional reporting year filter (applies an OCC_YEAR = <year> WHERE clause when offline = FALSE).
max_features	Optional cap on returned rows (offline = FALSE only).
offline	If TRUE (default), read the bundled synthetic fixture from inst/extdata/tps_mha_apprehension (5 rows in the canonical TPS PSDP 22-column schema). If FALSE, hit the TPS PSDP ArcGIS FeatureServer.
layer_url	Optional ArcGIS layer URL override.

`morie_datasets_tps_investigated_alleged_misconduct`
Investigated Alleged Misconduct (ASR-PCF-TBL-002)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `aaea16d94ae64da8a790d9649788de4c`.

Usage

```
morie_datasets_tps_investigated_alleged_misconduct(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a breakdown of the total investigated complaints by type of complaint submitted.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Misconduct

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_killed_and_seriously_injured`
Killed and Seriously Injured

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `0a1ee9d9436546dcbdc0ee9301e45e83`.

Usage

```
morie_datasets_tps_killed_and_seriously_injured(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Killed or Seriously Injured (KSI) - related collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; MVC; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_layers`
List the TPS open-data layers bundled with morie.

Description

List the TPS open-data layers bundled with morie.

Usage

```
morie_datasets_tps_layers()
```

Value

A `data.frame` with columns `name` and `url`.

`morie_datasets_tps_major_crime`
TPS Major Crime Indicators feed.

Description

TPS Major Crime Indicators feed.

Usage

```
morie_datasets_tps_major_crime(
  year = NULL,
  max_features = NULL,
  include_geometry = FALSE,
  offline = FALSE
)
```

Arguments

`year`	Integer or `NULL`. If set, filter to `OCC_YEAR == year` server-side.
`max_features`	Integer or `NULL`; cap on returned rows.
`include_geometry`	Logical; include `geom_x` / `geom_y`.
`offline`	Logical; return the bundled synthetic frame instead of hitting the live ArcGIS endpoint.

Value

A `data.frame` with the documented TPS schema.

`morie_datasets_tps_mha_apprehensions` *TPS Mental Health Act Apprehensions (PSDP)*

Description

Wraps the TPS Public Safety Data Portal "Mental Health Act Apprehensions" layer (one row per police-attended MHA event). Carries both HOOD_158 and HOOD_140 columns – callers should pick a version via `morie_tps_resolve_hood_col()` before downstream analysis.

Usage

```
morie_datasets_tps_mha_apprehensions(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

`year`	Optional reporting year filter (applies an `OCC_YEAR = <year>` WHERE clause when `offline = FALSE`).
`max_features`	Optional cap on returned rows (`offline = FALSE` only).
`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/tps_mha_apprehension` (5 rows in the canonical TPS PSDP 22-column schema). If `FALSE`, hit the TPS PSDP ArcGIS FeatureServer.
`layer_url`	Optional ArcGIS layer URL override.

Value

A `data.frame`.

References

TPS Public Safety Data Portal, "Mental Health Act Apprehensions Open Data" (https://data.tps.ca/datasets/333c4e1c96314741a83425045b6a7642_0/explore).

Examples

```
df <- morie_datasets_tps_mha_apprehensions(offline = TRUE)
table(df$APPREHENSION_TYPE)
```

`morie_datasets_tps_miscellaneous_calls_for_service`
Miscellaneous Calls for Service (ASR-CS-TBL-002)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `542374a83ea64b3ba222c41309445b8e`.

Usage

```
morie_datasets_tps_miscellaneous_calls_for_service(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset includes the following categories of data: Languages, Calls Received, and Alarm Calls

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Calls for Service

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_miscellaneous_data` *Miscellaneous Data (ASR-MISC-TBL-001)*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `bc229f576f174e24946bd1649c98aa43`.

Usage

```
morie_datasets_tps_miscellaneous_data(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset contains information pertaining to intimate partner violence, hate crimes and budget.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Hate Crime; Budget; IPV

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_miscellaneous_firearms`
Miscellaneous Firearms (ASR-F-TBL-003)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `7f9e4439a6e749dea32dab9e1704b58a`.

Usage

```
morie_datasets_tps_miscellaneous_firearms(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Firearms

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_motorcylist_ksi` *Motorcylist KSI*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `d691a9391c2a4c6d85bb761530d33310`.

Usage

```
morie_datasets_tps_motorcylist_ksi(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Motorcyclist-related KSI collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; Motorcyclist; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_neighbourhood_crime_rates` *Neighbourhood Crime Rates Open Data*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `ea0cfecdb1de416884e6b0bf08a9e195`.

Usage

```
morie_datasets_tps_neighbourhood_crime_rates(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; ”offline” here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Neighbourhood crime rates per 100,000.

Tags: Neighbourhood; Crime; Rate; Crime Rates; Community Safety Indicators; CSI; Toronto; TPS; Toronto Police

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_passenger_ksi` *Passenger KSI*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `e4e28a899191479d8e53754414894870`.

Usage

```
morie_datasets_tps_passenger_ksi(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Passenger-related KSI collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; Passenger; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_patrol_zone` *Patrol Zone*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `4a02ac3ed83d478c914d62c3064d6bc4`.

Usage

```
morie_datasets_tps_patrol_zone(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Police patrol zones.

Tags: Patrol Zone; Boundary Files; Patrol Zones

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_pedestrian_ksi` *Pedestrian KSI*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `a96252bf61b84fc68c3926bb7485970e`.

Usage

```
morie_datasets_tps_pedestrian_ksi(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Pedestrian-related KSI collisions (2006 - 2023).

Tags: Killed or Seriously Injured; KSI; Pedestrian; Traffic; Collision; Traffic Collision

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_personnel_by_command`
Personnel by Command (ASR-PB-TBL-004)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `1f58109772e2484fba0f509c1ab49fe8`.

Usage

```
morie_datasets_tps_personnel_by_command(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a count of personnel broken down by command level.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Personnel

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_personnel_by_rank` *Personnel by Rank (ASR-PB-TBL-002)*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `62016275c866412d8de5db539dc0cb8a`.

Usage

```
morie_datasets_tps_personnel_by_rank(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a count of personnel broken down by rank classification for Uniform, Civilian, and Other Staff.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Personnel

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_personnel_by_rank_by_division`
Personnel by Rank by Division (ASR-PB-TBL-003)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `e29b8d05c4754b3b8fc234324811a897`.

Usage

```
morie_datasets_tps_personnel_by_rank_by_division(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a count of personnel broken down by rank classification for Uniform & Civilian staff by division. This data includes the command level at the time of reporting.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Personnel

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_persons_in_crisis_calls_for_service_attended`
Persons in Crisis Calls for Service Attended Open Data

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `79c8e950bfe54ce39334ba108e1b325f`.

Usage

```
morie_datasets_tps_persons_in_crisis_calls_for_service_attended(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; ”offline” here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Persons in crisis calls for service attended.

Tags: Persons in Crisis; PIC; Crisis; Apprehensions; MHA; Calls; Calls for Service; Toronto; TPS; Toronto Police

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_police_divisions` *Police Divisions*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `fda21b25213c4c07b08c5162cba5081f`.

Phase 3CCC3. Bundled snapshot of the TPS Hub `fda21b25213c4c07b08c5162cba5081f` (TPS_POLICE_DIVISIONS) – the 16 post-amalgamation TPS divisions (D11, D12, D13, D14, D22, D23, D31, D32, D33, D41, D42, D43, D51, D52, D53, D55) with unit name, address, and area_sqkm.

Usage

```
morie_datasets_tps_police_divisions(offline = TRUE, max_features = NULL)

morie_datasets_tps_police_divisions(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV.
`max_features`	Optional row cap.

Details

Police divisions (post D54/D55 amalgamation).

Tags: City of Toronto; Toronto; Open Data; Feature Class; Update; Data Load; Divisions; Police Divisions

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

A `data.frame` with `DIV`, `UNIT_NAME`, `ADDRESS`, `CITY`, `AREA_SQKM`, plus shape area / perimeter fields.

Examples

```
df <- morie_datasets_tps_police_divisions(offline = TRUE)
nrow(df)  # 16
```

`morie_datasets_tps_psdp_resolved` *One-call TPS PSDP data + boundary metadata join*

Description

Phase 3CCC3. Pulls a TPS PSDP crime dataset (`morie_datasets_tps_assault()` etc.) and left-joins its native DIVISION + HOOD_158 + HOOD_140 columns against the bundled boundary metadata loaders (`morie_datasets_tps_police_divisions()`, `morie_to_neighbourhoods()` 158 + 140 + NIA flags) and the PSDP layer registry (`morie_tps_psdp_layers()`).

Usage

```
morie_datasets_tps_psdp_resolved(
  layer_key,
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL,
  resolvers = c("division", "hood158", "hood140", "nia", "psdp_class")
)
```

Arguments

`layer_key`	One of the PSDP `layer_keys` from `morie_tps_psdp_layers()` (e.g., `"assault"`, `"autotheft"`, `"homicides"`).
`year`	Optional year filter passed through to the underlying loader.
`max_features`	Optional row cap.
`offline`	If `TRUE` (default), all data come from bundled fixtures.
`layer_url`	Backward-compat override for non-canonical FeatureServer URL.
`resolvers`	Character subset of `c("division", "hood158", "hood140", "nia", "psdp_class")`. Default joins all five.

Details

Per-row PSDP datasets carry these ID columns natively:

- `DIVISION` (e.g., "D11") – joins to police_divisions.DIV
- `HOOD_158` (integer 1-158) – joins to 158-neighbourhoods
- `NEIGHBOURHOOD_158` (denormalised name)
- `HOOD_140` (integer 1-140) – joins to 140-neighbourhoods
- `NEIGHBOURHOOD_140` (denormalised name)

Resolver columns are prefixed `division_*`, `hood158_*`, `hood140_*`, `nia_*`, `psdp_*` to avoid collisions. Left-join semantics (row count preserved).

Mirrors the Chicago `morie_datasets_chicago_crime_resolved()` (3VV+) and NYPD `morie_datasets_nyc_nypd_resolved()` (3AAA-3CCC1) patterns.

Value

A wide `data.frame`: PSDP columns first, then prefixed resolver columns.

Examples

```
df <- morie_datasets_tps_psdp_resolved("assault", offline = TRUE)
names(df)
```

---

morie_datasets_tps_regulated_interactions

Regulated Interactions (ASR-RI-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `1cd5d478ef79424a8a6d5319a44edb0a`.

Usage

```
morie_datasets_tps_regulated_interactions(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

The data provided count describes situations involving regulated interactions.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Regulated Interactions

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_reported_crimes`
Reported Crimes (ASR-RC-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `fe2e40a464e64cb3a0e69ac3ccd17dfa`.

Usage

```
morie_datasets_tps_reported_crimes(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset includes all reported crime offences by reported date aggregated by division.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Reported Crimes

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_robbery`
TPS PSDP – Robbery

Description

TPS PSDP – Robbery

Usage

```
morie_datasets_tps_robbery(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

`year`	Optional reporting year filter (applies an `OCC_YEAR = <year>` WHERE clause when `offline = FALSE`).
`max_features`	Optional cap on returned rows (`offline = FALSE` only).
`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/tps_mha_apprehension` (5 rows in the canonical TPS PSDP 22-column schema). If `FALSE`, hit the TPS PSDP ArcGIS FeatureServer.
`layer_url`	Optional ArcGIS layer URL override.

`morie_datasets_tps_search_of_persons`
Search of Persons (ASR-SP-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `8ee1697ce6af44a78640686a1feeeefb`.

Usage

```
morie_datasets_tps_search_of_persons(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset includes all Level 3 and Level 4 searches that were conducted.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Search of Person

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_shooting_firearm_discharges`

TPS PSDP – Shooting and Firearm Discharges

Description

TPS PSDP – Shooting and Firearm Discharges

Usage

```
morie_datasets_tps_shooting_firearm_discharges(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

year	Optional reporting year filter (applies an OCC_YEAR = <year> WHERE clause when offline = FALSE).
max_features	Optional cap on returned rows (offline = FALSE only).
offline	If TRUE (default), read the bundled synthetic fixture from inst/extdata/tps_mha_apprehension (5 rows in the canonical TPS PSDP 22-column schema). If FALSE, hit the TPS PSDP ArcGIS FeatureServer.
layer_url	Optional ArcGIS layer URL override.

---

morie_datasets_tps_shootings

TPS Shootings and Firearm Discharges feed.

Description

TPS Shootings and Firearm Discharges feed.

Usage

```
morie_datasets_tps_shootings(year = NULL, max_features = NULL)
```

Arguments

year	Integer or NULL. If set, filter to OCC_YEAR == year server-side.
max_features	Integer or NULL; cap on returned rows.

Value

A data.frame.

`morie_datasets_tps_staffing_by_command` *Staffing_by_Command*

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `9d97ef7e8b494095be4abc0a628d7ce3`.

Usage

```
morie_datasets_tps_staffing_by_command(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Toronto Police Service dataset.

Tags: Staffing; Budget; Toronto Police Service; TPS

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_theft_from_motor_vehicle`
TPS PSDP – Theft From Motor Vehicle

Description

TPS PSDP – Theft From Motor Vehicle

Usage

```
morie_datasets_tps_theft_from_motor_vehicle(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

`year`	Optional reporting year filter (applies an `OCC_YEAR = <year>` WHERE clause when `offline = FALSE`).
`max_features`	Optional cap on returned rows (`offline = FALSE` only).
`offline`	If `TRUE` (default), read the bundled synthetic fixture from `inst/extdata/tps_mha_apprehension` (5 rows in the canonical TPS PSDP 22-column schema). If `FALSE`, hit the TPS PSDP ArcGIS FeatureServer.
`layer_url`	Optional ArcGIS layer URL override.

`morie_datasets_tps_theft_over`
TPS PSDP – Theft Over

Description

TPS PSDP – Theft Over

Usage

```
morie_datasets_tps_theft_over(
  year = NULL,
  max_features = NULL,
  offline = TRUE,
  layer_url = NULL
)
```

Arguments

year	Optional reporting year filter (applies an OCC_YEAR = <year> WHERE clause when offline = FALSE).
max_features	Optional cap on returned rows (offline = FALSE only).
offline	If TRUE (default), read the bundled synthetic fixture from inst/extdata/tps_mha_apprehension (5 rows in the canonical TPS PSDP 22-column schema). If FALSE, hit the TPS PSDP ArcGIS FeatureServer.
layer_url	Optional ArcGIS layer URL override.

---

morie_datasets_tps_tickets_issued

Tickets Issued (ASR-ENF-TBL-002)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to morie_datasets_tps_arcgis_hub_by_id() with the canonical hub item_id 5069c21b5b364194807bf1958556b1ff.

Usage

```
morie_datasets_tps_tickets_issued(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

format	One of "json" (default), "geojson", "csv", "shapefile", "fgdb".
where	SoQL-style WHERE filter passed to FeatureServer ?where= (default "1=1").
max_features	Optional cap on returned rows. Passed as resultRecordCount to FeatureServer; ignored for binary formats.
layer_idx	Integer index of the FeatureServer layer to pull (default 0L, the first layer).
offline	Logical; if TRUE, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default TRUE – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
dest	Optional path for binary downloads (format %in% c("shapefile", "fgdb")); defaults to tempfile().

Details

This dataset provides an aggregated count of tickets issued by year, ticket type, offence, age group, division, and neighbourhood.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Tickets

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_top_20_offences_of_firearm_seizures`

Top 20 Offences of Firearm Seizures (ASR-F-TBL-002)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `a83aa604fed240acaf2dfe64e1b323f8`.

Usage

```
morie_datasets_tps_top_20_offences_of_firearm_seizures(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a list of top 20 offences ranked by volume, for occurrences linked to a firearm seizure.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Firearms

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_total_public_complaints`

Total Public Complaints (ASR-PCF-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `a16edf4bc9484e94ad7e00bc22727544`.

Usage

```
morie_datasets_tps_total_public_complaints(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset provides a breakdown of the total number of public complaints from the Law Enforcement Complaints Agency (L.E.C.A.) broken down by complaints that were investigated and not investigated.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Complaints

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_traffic_collisions`

Traffic Collisions Open Data (ASR-T-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `bc4c72a793014a55a674984ef175a6f3`.

Usage

```
morie_datasets_tps_traffic_collisions(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

Collision occurrences by occurrence date.

Tags: Traffic; Collision; Traffic Collisions; Motor Vehicle Collisions; Toronto; TPS; Toronto Police

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_call_for_service_types`

Use of Force: Call for Service Types (RBDC-UOF-TBL-004)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `04633ebdaba941efaa82f2cdaaa00bb8`.

Usage

```
morie_datasets_tps_use_of_force_call_for_service_types(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides information about the types of calls for service which resulted in an enforcement action and/or reported use of force.

Tags: race; race based data; RBDC

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_call_sources_by_month`

Use of Force: Call Sources by Month (RBDC-UOF-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `98d88b18c0364c8d86e6a7c690037b85`.

Usage

```
morie_datasets_tps_use_of_force_call_sources_by_month(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides monthly counts of incidents which officers responded to from different call sources and which resulted in an enforcement action and/or reported use of force.

Tags: RBDC; race based data; race

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_gender_composition`

Use of Force: Gender Composition (RBDC-UOF-TBL-006)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `de9284945c3e479e938c4b77586535b1`.

Usage

```
morie_datasets_tps_use_of_force_gender_composition(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides information about the genders of people involved in enforcement action incidents, including those that may be associated with a reported use of force.

Tags: race; race based data; rbdc

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_location_of_occurrences`
Use of Force: Location of Occurrences (RBDC-UOF-TBL-003)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `0e7f95cb45704c8e8c9a05973422211c`.

Usage

```
morie_datasets_tps_use_of_force_location_of_occurrences(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides location information, aggregated to the division level.

Tags: RBDC; race; race based data

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_occurrence_category`
Use of Force: Occurrence Category (RBDC-UOF-TBL-005)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `a9b6bef1d34b44eea814e1869fdcda62`.

Usage

```
morie_datasets_tps_use_of_force_occurrence_category(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides information about the nature of the incident or the most serious offence associated with the incident, after officers arrive to the scene.

Tags: rbdc; race; race based data

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_time_of_day_trends`
Use of Force: Time of Day Trends (RBDC-UOF-TBL-002)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `b2bd6427e19a4706a4727338824a82b6`.

Usage

```
morie_datasets_tps_use_of_force_time_of_day_trends(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides information on the time periods of the day in which incidents took place and which resulted in an enforcement action and/or reported use of force.

Tags: RBDC; race; race based data

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_use_of_force_use_of_force_types_and_perceived_weapons`

Use of Force: Use of Force Types and Perceived Weapons (RBDC-UOF-TBL-007)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `9388798a44cd4ee5bc175669d8b6fb13`.

Usage

```
morie_datasets_tps_use_of_force_use_of_force_types_and_perceived_weapons(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This table provides information about reported use of force incidents and the highest type of force used as well as whether officers perceived weapons were carried by people involved.

Tags: race; race based data; rbdc

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_tps_victims_of_crime`

Victims of Crime (ASR-VC-TBL-001)

Description

Toronto PS ArcGIS Hub dataset wrapper. Thin dispatch to `morie_datasets_tps_arcgis_hub_by_id()` with the canonical hub item_id `6afabfd5109847a2bbba3eaeb0275e35`.

Usage

```
morie_datasets_tps_victims_of_crime(
  format = "json",
  where = "1=1",
  max_features = NULL,
  layer_idx = 0L,
  offline = TRUE,
  dest = NULL
)
```

Arguments

`format`	One of `"json"` (default), `"geojson"`, `"csv"`, `"shapefile"`, `"fgdb"`.
`where`	SoQL-style `WHERE` filter passed to FeatureServer `?where=` (default `"1=1"`).
`max_features`	Optional cap on returned rows. Passed as `resultRecordCount` to FeatureServer; ignored for binary formats.
`layer_idx`	Integer index of the FeatureServer layer to pull (default `0L`, the first layer).
`offline`	Logical; if `TRUE`, the hub_id is resolved via the bundled catalog (no network needed for the resolution step). Default `TRUE` – you can run this against the 71 catalog entries without network. Live data fetches always hit the network regardless of this argument; "offline" here only affects the hub_id -> FeatureServer URL lookup.
`dest`	Optional path for binary downloads (`format %in% c("shapefile", "fgdb")`); defaults to `tempfile()`.

Details

This dataset includes all identified victims of crimes against the person, including, but not limited to, those that may have been deemed unfounded after investigation, those that may have occurred outside the City of Toronto limits, or have no verified location.

Tags: ASR; TPS; Toronto Police; Annual Statistical Report; Victims

Value

A data.frame / GeoJSON list / file path; see `morie_datasets_tps_arcgis_hub_by_id()` for per-format semantics.

`morie_datasets_vancouver_opendata_by_id`

Fetch records from a Vancouver Open Data dataset by ID

Description

Phase 3CCC4. Hits the Opendatasoft v2.1 `/records` endpoint for an arbitrary Vancouver dataset slug. Returns the `results` array as a `data.frame`. For larger pulls, use `format = "csv"` to hit the unrestricted `/exports/csv` endpoint instead.

Usage

```
morie_datasets_vancouver_opendata_by_id(
  dataset_id,
  limit = 100L,
  format = c("json", "csv")
)
```

Arguments

`dataset_id`	Opendatasoft dataset slug (from `morie_datasets_vancouver_opendata_layers()`).
`limit`	Page size (max 100 for `/records`).
`format`	One of `"json"` (default, `/records` endpoint) or `"csv"` (`/exports/csv` endpoint, no row limit).

Value

A `data.frame` of records.

Examples

```
## Not run:
df <- morie_datasets_vancouver_opendata_by_id("non-market-housing",
                                              limit = 50)
nrow(df)

## End(Not run)
```

`morie_datasets_vancouver_opendata_layers`
Vancouver Open Data full dataset catalog (Opendatasoft v2.1)

Description

Phase 3CCC4. Bundled snapshot of every City-of-Vancouver dataset published on opendata.vancouver.ca (190 datasets as of 2026-05-24). Each row identifies a dataset by its Opendatasoft `dataset_id` slug (used as the URL path segment for records / exports endpoints).

Usage

```
morie_datasets_vancouver_opendata_layers(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV; if `FALSE`, paginates the live catalog endpoint.
`max_features`	Optional row cap.

Value

A `data.frame` with `dataset_id`, `title`, `publisher`, `records_count`.

References

Opendatasoft Explore API v2.1, https://opendata.vancouver.ca/api-console/explore/v2.1/.

Examples

```
cat_df <- morie_datasets_vancouver_opendata_layers(offline = TRUE)
nrow(cat_df)  # 190
head(cat_df$title)
```

`morie_datasets_vpd_crime`
Load Vancouver Police Department crime incident data

Description

Phase 3DDD2. Loads VPD's open crime incident records. Three source modes:

Usage

```
morie_datasets_vpd_crime(
  offline = TRUE,
  zip_path = NULL,
  csv_path = NULL,
  max_features = NULL,
  accept_terms = FALSE
)
```

Arguments

`offline`	If `TRUE` (default) and `zip_path`/`csv_path` are `NULL`, reads the bundled 550-row sample.
`zip_path`	Optional path to a user-downloaded `crimedata_csv_AllNeighbourhoods_AllYears.zip`. Mutually exclusive with `csv_path`.
`csv_path`	Optional path to a pre-extracted CSV. Mutually exclusive with `zip_path`.
`max_features`	Optional row cap.
`accept_terms`	If reading from a user-supplied zip/csv, pass `TRUE` to silently acknowledge VPD's terms (else a warning surfaces them once per session).

Details

`offline = TRUE` **(default)** Reads a bundled stratified 550-row sample (50 rows per `TYPE` x 11 categories) covering years 2003-2026 and all 25 VPD-defined neighbourhoods. Intended for tests + intro examples – NOT for analysis.

`zip_path = "..."` Reads from a local copy of VPD's `crimedata_csv_AllNeighbourhoods_AllYears.zip` that the caller has downloaded themselves from https://geodash.vpd.ca/opendata/ (after accepting VPD's terms + conditions there).

`csv_path = "..."` Reads from a pre-extracted CSV (skip the zip if the caller already has the CSV on disk).

The bundled sample is open-licensed under VPD's GeoDASH terms; the full feed requires manual T&C acceptance per VPD policy and there is no automation-friendly API. See `morie_datasets_vpd_legal_disclaimer()` for the full text.

Columns (10): `TYPE`, `YEAR`, `MONTH`, `DAY`, `HOUR`, `MINUTE`, `HUNDRED_BLOCK`, `NEIGHBOURHOOD`, `X`, `Y`. Coordinates are UTM Zone 10 N (NAD83 / EPSG:26910). For Offence-Against-a-Person incidents the location is deliberately randomized + offset per VPD's privacy policy.

Categories present in the sample:

- Break and Enter Commercial
- Break and Enter Residential/Other
- Homicide
- Mischief
- Offence Against a Person (aggregated)
- Other Theft

- Theft from Vehicle
- Theft of Bicycle
- Theft of Vehicle
- Vehicle Collision or Pedestrian Struck (with Fatality)
- Vehicle Collision or Pedestrian Struck (with Injury)

Value

A `data.frame` with 10 columns.

Data quality + interpretation caveats (per VPD GeoDASH disclaimer)

- **Source**: extracted from the PRIME-BC Police Records Management System (RMS); filtered + aggregated to comply with the BC Freedom of Information & Protection of Privacy Act (BC FIPPA).
- `Offence Against a Person` **is INTENTIONALLY aggregated** to reduce re-identification risk. It bundles robbery, assault (incl. sexual assault, domestic assault), and other violent incidents EXCEPT `Assaults Against Police`. Sub-categories are deliberately NOT exposed; do not attempt to disaggregate this column.
- `Other Theft` **aggregates** shoplifting, theft of personal property (over / under $5000), mail theft, and utilities theft.
- **Reporting method**: 'All Offence' + 'Founded' (incidents the investigating officer determined did occur). This is **NOT comparable** to Statistics Canada's published numbers, which use 'UCR Survey' Most-Serious-Offence (MSO) scoring. Do not mix VPD GeoDASH totals with StatCan totals in the same denominator.
- **Location precision is deliberately reduced**: person-crimes have their X/Y **randomized to several blocks and offset to an intersection**; no time/street-name is provided. Property-crimes are provided at the **hundred-block** level only. Never interpret a row's X/Y as the actual scene of the incident.
- **Crime classification + file status may change retroactively** as investigations evolve. The dataset is a snapshot, not an archive of fact.
- **Update schedule**: VPD refreshes the feed every Sunday morning. Cache locally for reproducible analysis.
- **Not a calls-for-service log**: only incidents that passed the founded-categorization filter appear. Totals do **not** reflect total calls or complaints made to the VPD.
- **Liability disclaimer**: VPD / Vancouver Police Board / City of Vancouver assume no liability for any decision made from this data. morie surfaces it as-is.

References

VPD GeoDASH Open Data, https://geodash.vpd.ca/opendata/.

Examples

```
df <- morie_datasets_vpd_crime(offline = TRUE)
nrow(df)              # 550
table(df$TYPE)
table(df$NEIGHBOURHOOD)
```

`morie_datasets_vpd_legal_disclaimer`
Read VPD's legal disclaimer (bundled verbatim from the zip)

Description

Phase 3DDD2. Returns the legal disclaimer text shipped with VPD's open crime data download. Useful in headless or programmatic workflows where the human reader of the GeoDASH web UI is not the same as the script user.

Usage

```
morie_datasets_vpd_legal_disclaimer()
```

Value

A character vector (one element per line).

`morie_db_connect` *Connect to the MORIE cache database*

Description

Opens (or creates) the per-user cache database. The default backend is **DuckDB** — zero-config like SQLite, but vectorised + columnar, so it handles the multi-GB-scale open-data PUMFs (TPS, CPADS bulk) that morie ingests without breaking down on analytical queries. For back-compat, an existing SQLite cache at `morie.db` is reused; if duckdb is unavailable, falls back to SQLite.

Usage

```
morie_db_connect(db_path = NULL)
```

Arguments

`db_path`	Optional path to a DuckDB (`*.duckdb`) or SQLite (`*.db`) file. Defaults to the `MORIE_CACHE_DB` env var, else `morie.duckdb` / `morie.db` in the per-user cache directory.

Details

For non-default backends (PostgreSQL, MariaDB, MS SQL Server, ...), construct your own DBI connection and pass it as `con` to the `morie_cache_*` and `morie_load_dataset` functions:

```
con <- DBI::dbConnect(RPostgres::Postgres(),
  host = "...", dbname = "morie", user = "...", password = "...")
morie_load_dataset("ocp21", con = con)
```

Value

A DBI connection object.

Examples

```
# DuckDB (default when 'duckdb' is installed); pass a '.db' path for SQLite.
if (requireNamespace("duckdb", quietly = TRUE) &&
  requireNamespace("DBI", quietly = TRUE)) {
  tmp <- tempfile(fileext = ".duckdb")
  con <- morie_db_connect(db_path = tmp)
  DBI::dbListTables(con)
  DBI::dbDisconnect(con)
  file.remove(tmp)
}
```

---

morie_db_create_indexes

Create the recommended B-tree indexes for a morie cache table

---

Description

Looks up `table_name` in the per-dataset index registry (see `.morie_db_index_registry()` for the full list) and creates each `CREATE INDEX IF NOT EXISTS` against `con`. Specs whose columns aren't present in the actual table are silently skipped, so this is safe to call on any morie cache table — including subsets that drop some columns. Returns the number of `CREATE INDEX` statements that actually ran (not the number registered).

Usage

```
morie_db_create_indexes(con, table_name)
```

Arguments

`con`	A DBI connection.
`table_name`	The cache table name (case-sensitive). Common short names: `"SIU"`, `"b01"`, `"c01"`, `"d01"`, `"uof_main_records"`, `"assault"`, `"homicide"`, etc.

Details

Cardinality-based selection: every indexed column has > 30 distinct values in the real published data; low-cardinality columns (`Gender`, `Yes`/`No` alerts, `Measure`) are intentionally skipped because the index overhead exceeds the lookup benefit.

Value

Invisibly returns the integer count of `CREATE INDEX` statements executed.

`morie_dbscan_clustering`
DBSCAN density-based clustering (R parity)

Description

Wraps `dbscan::dbscan`.

Usage

```
morie_dbscan_clustering(x, eps = 0.5, min_samples = 5L, metric = "euclidean")
```

Arguments

`x`	Numeric matrix.
`eps`	Neighbourhood radius.
`min_samples`	Minimum points in eps-neighbourhood for a core point.
`metric`	Distance metric (passed to dbscan).

Value

Named list: estimate, labels, n_clusters, n_noise, core_sample_indices, eps, min_samples, n, method.

Examples

```
morie_dbscan_clustering(x = rnorm(50))
```

`morie_dcc_multivariate_garch`
DCC multivariate GARCH (Engle 2002)

Description

Two-step DCC(1,1) on a panel of return series.

Usage

```
morie_dcc_multivariate_garch(x)
```

Arguments

`x`	Numeric matrix of returns (T x k).

Value

Named list with `a`, `b`, `unconditional_correlation`, `conditional_correlation`, `conditional_variance`, `loglik`, `n`, `k`, `method`.

Examples

```
morie_dcc_multivariate_garch(x = matrix(rnorm(150), 50, 3))
```

`morie_decision_tree_split`

Decision tree split (R parity)

Description

CART tree via `rpart::rpart`, returning the root split structure and feature importances.

Usage

```
morie_decision_tree_split(x, y, criterion = "gini", max_depth = 30L, seed = 0L)
```

Arguments

`x`	Numeric predictor matrix.
`y`	Response (factor for classification).
`criterion`	"gini" or "entropy" – only "gini" is supported by rpart for classification; "entropy" maps to information.
`max_depth`	Max tree depth.
`seed`	RNG seed.

Value

Named list: estimate, train_accuracy, root_feature, root_threshold, root_impurity, n_leaves, feature_importances, criterion, n, method.

Examples

```
morie_decision_tree_split(x = rnorm(50), y = rnorm(50))
```

`morie_deep_learning_genomic` *Single-hidden-layer MLP genomic predictor (base R)*

Description

Single-hidden-layer MLP genomic predictor (base R)

Usage

```
morie_deep_learning_genomic(
  x,
  y,
  markers,
  hidden = 16,
  n_epochs = 200,
  lr = 0.01,
  l2 = 0.001,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	Fixed-effect design (optional).
`y`	Numeric response.
`markers`	Genotype matrix (n x m).
`hidden`	Hidden units (default 16).
`n_epochs`	Training epochs.
`lr`	Learning rate.
`l2`	L2 weight decay.
`seed`	Seed.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("dlgen", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, y_hat, beta, W1, b1, w2, b2, se, n, method).

References

Montesinos Lopez Ch 12.

Examples

```
morie_deep_learning_genomic(
  x = rnorm(50), y = rnorm(50),
  markers = matrix(sample(0:2, 200, TRUE), 50, 4)
)
```

---

morie_default_synthetic_name_map

Default synthetic-data variable name map

Description

Returns a named character vector mapping canonical variable keys used by `morie_generate_synthetic_data()` to output column names.

Usage

```
morie_default_synthetic_name_map(profile = c("generic", "morie_legacy"))
```

Arguments

`profile`	Name profile. `"generic"` is recommended for new projects. `"morie_legacy"` reproduces previous EML legacy column names.

Value

Named character vector.

Examples

```
morie_default_synthetic_name_map("generic")
```

---

morie_default_workflow_map

Default workflow step map

Description

Returns the default named map of workflow steps to project script paths.

Usage

```
morie_default_workflow_map()
```

Value

Named character vector.

Examples

```
morie_default_workflow_map()
```

---

morie_desc_atkinson *Atkinson inequality index via* **DescTools**

Description

Thin extender over `DescTools::Atkinson` for the inequality-aversion-parameter family of inequality indices.

Usage

```
morie_desc_atkinson(x, parameter = 0.5, ...)
```

Arguments

`x`	A non-negative numeric vector.
`parameter`	Inequality-aversion parameter (default 0.5), forwarded to `DescTools::Atkinson`.
`...`	Further arguments forwarded to `DescTools::Atkinson` (e.g. `na.rm`).

Value

A list with `$method = "DescTools::Atkinson"` and `$raw` (the Atkinson index).

---

morie_desc_cramers_v *Cramer's V via* **DescTools**

Description

Thin extender over `DescTools::CramerV` for the symmetric association statistic between two categorical variables.

Usage

```
morie_desc_cramers_v(x, y = NULL, ...)
```

Arguments

`x`	A factor, character vector, or contingency table.
`y`	Optional second categorical vector when `x` is not already a contingency table.
`...`	Further arguments forwarded to `DescTools::CramerV` (e.g. `conf.level`).

Value

A list with `$method = "DescTools::CramerV"` and `$raw` (the upstream numeric or matrix).

`morie_desc_gini` *Gini coefficient via* **DescTools**

Description

Thin extender over `DescTools::Gini` for the Gini index of (in)equality on a non-negative numeric vector.

Usage

```
morie_desc_gini(x, ...)
```

Arguments

`x`	A non-negative numeric vector.
`...`	Further arguments forwarded to `DescTools::Gini` (e.g. `weights`, `unbiased`, `conf.level`, `na.rm`).

Value

A list with `$method = "DescTools::Gini"` and `$raw` (the Gini estimate, optionally with CI).

`morie_desc_kappa` *Cohen / Fleiss kappa via* **DescTools**

Description

Thin extender over `DescTools::CohenKappa` (when `y` is supplied) or `DescTools::KappaM` (when `x` is a multi-rater matrix / data frame) for inter-rater agreement.

Usage

```
morie_desc_kappa(x, y = NULL, ...)
```

Arguments

`x`	A vector of ratings (paired with `y`) or a matrix / data frame whose columns are raters.
`y`	Optional second rater vector when `x` is a vector.
`...`	Further arguments forwarded to the upstream function.

Value

A list with `$method` (qualified upstream name) and `$raw` (the upstream return object).

morie_desc_winsorize *Winsorize a numeric vector via* **DescTools**

Description

Thin extender over `DescTools::Winsorize` for symmetric winsorisation by quantile (default 5\

Usage

```
morie_desc_winsorize(x, probs = c(0.05, 0.95), ...)
```

Arguments

`x`	A numeric vector.
`probs`	Length-2 numeric vector of lower / upper quantile probabilities, forwarded to `DescTools::Winsorize`.
`...`	Further arguments forwarded to `DescTools::Winsorize` (e.g. `minval`, `maxval`, `na.rm`).

Value

A list with `$method = "DescTools::Winsorize"` and `$raw` (the winsorised numeric vector).

morie_describe *Print the pedagogical narrative for a morie callable.*

Description

Loads the describe_.md narrative shipped in the package's `inst/extdata/describe_corpus.Rds` bundle and prints it to the console. This is the R-side mirror of the Python `morie.describe()` function (closing the v0.9.5.4 parity gap; shipped in v0.9.5.5).

Usage

```
morie_describe(callable)
```

Arguments

`callable`	A morie callable, as a function object (passed unquoted), or a character scalar name. The lookup strips the leading `morie_` prefix automatically.

Details

The lookup is forgiving: a leading `morie_` prefix on the callable name is stripped automatically, so `morie_describe("aalen")` and `morie_describe("morie_aalen")` resolve to the same narrative.

Value

Invisibly returns the narrative as a character scalar. If no matching describe entry is found, returns `NULL` and prints a helpful diagnostic.

See Also

`morie_describe_by_name` for the string-only variant that does not capture symbol names.

Examples

```
## Not run:
morie_describe("aalen")
morie_describe("morie_aalen")  # leading prefix stripped
morie_describe(morie_aalen)    # function-object form

## End(Not run)
```

---

morie_describe_by_name

String-only variant of `morie_describe`.

Description

Use this when you want to pass a name as a string and avoid the unquoted-symbol capture behaviour of `morie_describe`.

Usage

```
morie_describe_by_name(name)
```

Arguments

`name`	Character scalar, the callable's mnemonic name (with or without the `morie_` prefix).

Value

Invisibly returns the narrative as a character scalar. If no matching describe entry is found, returns `NULL` and prints a helpful diagnostic.

Examples

```
## Not run:
morie_describe_by_name("aalen")
morie_describe_by_name("morie_aalen")

## End(Not run)
```

morie_design_effect *Design effect (DEFF)*

Description

Design effect (DEFF)

Usage

```
morie_design_effect(weights)
```

Arguments

`weights`	Numeric vector of sampling weights.

Value

Numeric design effect (= n / Kish effective sample size).

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_det_rng *SHA-keyed deterministic RNG for Py<->R parity*

Description

Given a callable / fixture name and an integer seed, derive a stable R-side seed value via SHA-256, install it with `set.seed()`, and return it invisibly. The matched Python helper `morie._det_rng.from_seed(name, seed)` builds a `numpy.random.Generator` from the same SHA digest so bootstrap / MCMC draws on the two sides agree to Monte-Carlo tolerance (and bit-identical when a deterministic-pseudo-bootstrap mode is plumbed).

Usage

```
morie_det_rng(name, seed)
```

Arguments

`name`	Character scalar; stable callable / fixture name. Must be identical to the string the Python side passes.
`seed`	Integer; user-supplied seed.

Details

Mechanism: `SHA-256(paste0(name, ":", seed))` is truncated to 32 bytes; bytes `[9:12]` (1-indexed, i.e. hex chars 17..24) form a 32-bit value reduced modulo `2^31 - 1` and passed to `set.seed()`. Bytes `[1:8]` are reserved for the Python Philox key. See `inst/python-stub/det_rng.py` (or the parent `morie/_det_rng.py`) for the Python counterpart.

Requires either the `digest` or `openssl` package for SHA-256. Both are widely available on CRAN; we try `digest` first, then `openssl`, and finally fall back to an internal pure-R SHA-256 implementation loaded only when neither is available. In practice CRAN reverse dependencies of `morie` ship with at least one of the two.

Value

Integer seed installed via `set.seed()` (invisibly).

Examples

```
morie_det_rng("ksr07_bootstrap", 42L)
rnorm(5) # reproducible draws keyed by ("ksr07_bootstrap", 42)
```

---

morie_det_rng_sha_hex

SHA-256 hex digest of "name:seed" (for Py<->R cross-check)

Description

Helper exposed so testthat can assert the Python and R sides compute identical hex digests for the same `(name, seed)` pair before either RNG is even consulted.

Usage

```
morie_det_rng_sha_hex(name, seed)
```

Arguments

`name`	Character scalar.
`seed`	Integer scalar.

Value

64-character lowercase hex string.

Examples

```
morie_det_rng_sha_hex(name = "example", seed = 1L)
```

`morie_did_2x2` *Classic 2x2 Difference-in-Differences estimator*

Description

Estimates the canonical two-group / two-period DiD treatment effect

$$\hat{\tau} = (\bar{Y}_{1,\text{post}} - \bar{Y}_{1,\text{pre}}) - (\bar{Y}_{0,\text{post}} - \bar{Y}_{0,\text{pre}}).$$

With covariates, fits the regression $Y = \alpha + \beta D + \gamma P + \tau(D \times P) + X\delta + \varepsilon$ and reports $\hat{\tau}$.

Usage

```
morie_did_2x2(
  data,
  outcome,
  treatment,
  post,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Details

For multi-period staggered designs prefer `morie_did_group_time_att` (Callaway-Sant'Anna via **did**). `morie_did_doubly_robust` (via **DRDID**) is the recommended option when pre-treatment covariates are available.

Value

A list with elements `estimate`, `std_error`, `t_stat`, `p_value`, `ci_lower`, `ci_upper`, `n_treated`, `n_control`, `method`, `details`.

References

Angrist, J. D., & Pischke, J.-S. (2009). *Mostly Harmless Econometrics*. Princeton University Press.

Examples

```
## Not run:
df <- data.frame(
  y    = rnorm(200),
  d    = rep(c(0, 1), each = 100),
  post = rep(c(0, 1), times = 100)
)
morie_did_2x2(df, "y", "d", "post")

## End(Not run)
```

`morie_did_aggregate_gt_att` *Aggregate group-time ATTs into summary parameters*

Description

Mirrors the aggregation schemes available in `did::aggte` (overall ATT, by-cohort, by-calendar-time, by-event-time) but produces a tidy `data.frame` consumed by the rmorie / MRM downstream pipelines.

Usage

```
morie_did_aggregate_gt_att(
  gt_results,
  aggregation = "overall",
  time_col = "time",
  cohort_col = "cohort",
  att_col = "att",
  se_col = "std_error"
)
```

Arguments

`gt_results`	Output of `morie_did_group_time_att`.
`aggregation`	One of `"overall"` (default), `"cohort"`, `"calendar_time"`, `"event_time"`.
`time_col`, `cohort_col`, `att_col`, `se_col`	Column-name overrides.

Value

A data frame with `group`, `estimate`, `std_error`, `ci_lower`, `ci_upper`.

morie_did_bacon_decomposition

Goodman-Bacon decomposition of the TWFE DiD estimator

Description

Thin wrapper around `bacondecomp::bacon`. Decomposes a two-way fixed-effects DiD estimate into a weighted average of all possible 2x2 DiD comparisons. Hard-errors if **bacondecomp** is not installed.

Usage

```
morie_did_bacon_decomposition(data, outcome, treatment, unit, time)
```

Arguments

`data`	Balanced panel data.
`outcome`	Outcome column.
`treatment`	Binary treatment indicator that turns on at onset.
`unit`	Unit identifier.
`time`	Time period.

Value

A list with `components` (data frame) and `overall_estimate`.

References

Goodman-Bacon, A. (2021). Difference-in-differences with variation in treatment timing. *Journal of Econometrics*, 225(2), 254–277.

morie_did_chaisemartin_dhaultfoeuille

Heterogeneity-robust DiD (de Chaisemartin & D'Haultfoeuille, 2020)

Description

Thin wrapper around `DIDmultiplegt::did_multiplegt`. Computes the instantaneous treatment effect for switchers using appropriate comparisons. Hard-errors if **DIDmultiplegt** is not installed.

Usage

```
morie_did_chaisemartin_dhaultfoeuille(
  data,
  outcome,
  treatment,
  unit,
  time,
  n_bootstrap = 200L,
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	Panel data.
`outcome`, `treatment`, `unit`, `time`	Column names.
`n_bootstrap`	Bootstrap replications (forwarded as `brep`).
`seed`	RNG seed.
`alpha`	Significance level.

Value

A result list; see `morie_did_2x2`.

References

de Chaisemartin, C., & D'Haultfoeuille, X. (2020). Two-way fixed effects estimators with heterogeneous treatment effects. *American Economic Review*, 110(9), 2964–2996.

`morie_did_continuous_treatment`

DiD with a continuous (dose) treatment

Description

Estimates the marginal effect of a one-unit increase in treatment intensity in the post period.

Usage

```
morie_did_continuous_treatment(
  data,
  outcome,
  dose,
  post,
  covariates = NULL,
```

```
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

data	A data frame containing the outcome, treatment, post and any covariate columns.
outcome	Name of the outcome column.
dose	Continuous treatment-intensity column.
post	Name of the binary (0/1) post-period column.
covariates	Optional character vector of covariate column names.
cluster	Optional cluster ID column for CR1 standard errors.
alpha	Significance level for confidence intervals (default 0.05).

Value

A result list; see morie_did_2x2.

---

morie_did_diagnostics

Comprehensive diagnostics for a 2x2 DiD setting

Description

Reports group / period sample sizes, outcome distributions, and baseline covariate balance (standardised mean differences).

Usage

```
morie_did_diagnostics(
  data,
  outcome,
  treatment,
  post,
  covariates = NULL,
  cluster = NULL
)
```

Arguments

data	A data frame containing the outcome, treatment, post and any covariate columns.
outcome	Name of the outcome column.
treatment	Name of the binary (0/1) treatment-group column.
post	Name of the binary (0/1) post-period column.
covariates	Optional character vector of covariate column names.
cluster	Optional cluster ID column for CR1 standard errors.

Details

For richer covariate-balance reporting (variance ratios, KS statistics, love plots) prefer `cobalt::bal.tab` / `cobalt::love.plot`.

Value

A list with `sample_sizes`, `outcome_stats`, `covariate_balance`.

`morie_did_doubly_robust`
Doubly-robust DiD (Sant'Anna & Zhao, 2020)

Description

Thin wrapper around `DRDID::drdid_rc` for the 2x2 repeated-cross-section setting. Combines an outcome regression model with an inverse-probability weighting model and is consistent if either model is correctly specified. Hard-errors if **DRDID** is not installed.

Usage

```
morie_did_doubly_robust(
  data,
  outcome,
  treatment,
  post,
  covariates,
  ps_model = "logistic",
  or_model = "linear",
  cluster = NULL,
  n_bootstrap = 200L,
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`covariates`	Optional character vector of covariate column names.
`ps_model`	Unused; retained for back-compat. **DRDID** fits a logistic propensity-score model internally.
`or_model`	Unused; retained for back-compat. **DRDID** fits a linear outcome model internally.

`cluster`	Optional cluster ID column for CR1 standard errors.
`n_bootstrap`	Number of bootstrap replications (forwarded as `nboot`).
`seed`	RNG seed (set before the call).
`alpha`	Significance level for confidence intervals (default 0.05).

Details

For panel data (same units observed in both periods) prefer `DRDID::drdid_panel` directly.

Value

A result list; see `morie_did_2x2`.

References

Sant'Anna, P. H. C., & Zhao, J. (2020). Doubly robust difference-in-differences estimators. *Journal of Econometrics*, 219(1), 101–122.

`morie_did_event_study`

Event-study DiD specification

Description

Thin wrapper around `fixest::feols` with `fixest::i()` relative-time dummies, plus unit and time fixed effects. The `reference_period` is dropped as the baseline. Hard-errors if **fixest** is not installed.

Usage

```
morie_did_event_study(
  data,
  outcome,
  unit,
  time,
  treatment_time,
  covariates = NULL,
  reference_period = -1L,
  leads = 4L,
  lags = 4L,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Panel data frame.
`outcome`	Outcome column.
`unit`	Unit identifier column.
`time`	Calendar-time column (integer-valued).
`treatment_time`	Column giving the period in which each unit first received treatment (`Inf` or `NA` for never-treated units).
`covariates`	Optional time-varying covariates.
`reference_period`	Relative-time period omitted as baseline (default `-1`).
`leads`	Number of pre-treatment periods to include.
`lags`	Number of post-treatment periods to include.
`cluster`	Cluster variable for standard errors (defaults to `unit`).
`alpha`	Significance level.

Details

For sun-Abraham interaction-weighted estimation prefer `fixest::sunab()` directly.

Value

A list with `coefficients` (data frame), `reference_period`, `pre_trend_f_stat`, `pre_trend_p_value`, and `details`.

`morie_did_fuzzy` *Fuzzy DiD (LATE) via 2SLS*

Description

Uses $Z \times$ Post as an instrument for $D \times$ Post to recover a local average treatment effect under imperfect compliance.

Usage

```
morie_did_fuzzy(
  data,
  outcome,
  assignment,
  takeup,
  post,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`assignment`	Intent-to-treat assignment column.
`takeup`	Actual treatment-takeup column.
`post`	Name of the binary (0/1) post-period column.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Details

For the de Chaisemartin-D'Haultfoeuille fuzzy DiD estimator on panel data prefer `morie_did_chaisemartin_dhaul` (**DIDmultiplegt**).

Value

A result list; see `morie_did_2x2`.

`morie_did_group_time_att` *Callaway–Sant'Anna group-time average treatment effects*

Description

Thin wrapper around `did::att_gt`. For each cohort g and each post-treatment calendar period `t`, estimates $\mathrm{ATT}(g, t)$. Hard-errors if **did** is not installed.

Usage

```
morie_did_group_time_att(
  data,
  outcome,
  unit,
  time,
  treatment_time,
  covariates = NULL,
  method = "doubly_robust",
  control_group = "never_treated",
  n_bootstrap = 200L,
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	Panel data.
`outcome`	Outcome column.
`unit`	Unit identifier.
`time`	Calendar-time column (integer).
`treatment_time`	Column with treatment-onset period (use `Inf` for never-treated).
`covariates`	Optional covariates for doubly-robust estimation.
`method`	One of `"doubly_robust"` (default), `"ipw"`, or `"outcome_regression"`.
`control_group`	`"never_treated"` or `"not_yet_treated"`.
`n_bootstrap`	Number of bootstrap replications for inference (forwarded as `biters`).
`seed`	RNG seed (unused; retained for back-compat).
`alpha`	Significance level.

Value

A data frame with columns `cohort`, `time`, `att`, `std_error`, `ci_lower`, `ci_upper`, `p_value`.

References

Callaway, B., & Sant'Anna, P. H. C. (2021). Difference-in-Differences with multiple time periods. *Journal of Econometrics*, 225(2), 200–230.

`morie_did_heterogeneous` *Heterogeneity-robust DiD by sub-group / moderator quantile*

Description

Splits the sample by quantiles (or categories) of a moderator and estimates separate 2x2 DiDs.

Usage

```
morie_did_heterogeneous(
  data,
  outcome,
  treatment,
  post,
  moderator,
  covariates = NULL,
  cluster = NULL,
  n_quantiles = 4L,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`moderator`	Column to split on.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`n_quantiles`	Number of quantile bins if the moderator is continuous.
`alpha`	Significance level for confidence intervals (default 0.05).

Value

A data frame with one row per stratum.

`morie_did_panel_fe` *Two-way fixed-effects DiD (panel)*

Description

Thin wrapper around `fixest::feols` estimating $Y_{it} = \alpha_i + \lambda_t + \tau D_{it} + X'\delta + \varepsilon_{it}$ with cluster-robust standard errors. Hard-errors if **fixest** is not installed.

Usage

```
morie_did_panel_fe(
  data,
  outcome,
  treatment,
  unit,
  time,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`unit`	Unit identifier column.

`time`	Time period column.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Value

A result list; see `morie_did_2x2`.

`morie_did_parallel_trends_data` *Group-by-time outcome means for parallel-trends visualisation*

Description

Group-by-time outcome means for parallel-trends visualisation

Usage

```
morie_did_parallel_trends_data(data, outcome, treatment, time, weights = NULL)
```

Arguments

`data`	A data frame.
`outcome`, `treatment`, `time`	Column names.
`weights`	Optional survey weight column.

Value

A data frame with columns `time`, `group`, `mean_outcome`, `se`, `n`.

`morie_did_placebo_test_group` *Placebo DiD on sub-groups expected to be unaffected*

Description

Placebo DiD on sub-groups expected to be unaffected

Usage

```
morie_did_placebo_test_group(
  data,
  outcome,
  treatment,
  post,
  group_col,
  unaffected_groups,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`, `treatment`, `post`	Column names.
`group_col`	Column defining sub-groups.
`unaffected_groups`	Vector of group values where no effect is expected.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Value

A data frame, one row per placebo group.

`morie_did_placebo_test_outcome` *Placebo DiD on outcomes that should be unaffected*

Description

Placebo DiD on outcomes that should be unaffected

Usage

```
morie_did_placebo_test_outcome(
  data,
  placebo_outcomes,
  treatment,
  post,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`placebo_outcomes`	Character vector of outcome columns expected to show no treatment effect.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Value

A data frame, one row per placebo outcome.

`morie_did_placebo_test_time` *Placebo DiD at fake treatment times*

Description

For each candidate fake time in `placebo_times`, redefines the post indicator and estimates a 2x2 DiD on pre-true-treatment data.

Usage

```
morie_did_placebo_test_time(
  data,
  outcome,
  treatment,
  time,
  true_treatment_time,
  placebo_times,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`, `treatment`, `time`	Column names.
`true_treatment_time`	The actual treatment-onset time (data are restricted to pre-period observations).

placebo_times	Vector of candidate fake treatment times.
covariates	Optional character vector of covariate column names.
cluster	Optional cluster ID column for CR1 standard errors.
alpha	Significance level for confidence intervals (default 0.05).

Value

A data frame, one row per placebo time.

---

morie_did_repeated_cross_section

Repeated cross-section DiD (optionally weighted)

Description

Same specification as morie_did_2x2 but accepts a survey weight column. When weights is supplied, weighted least squares is used.

Usage

```
morie_did_repeated_cross_section(
  data,
  outcome,
  treatment,
  post,
  covariates = NULL,
  weights = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

data	A data frame containing the outcome, treatment, post and any covariate columns.
outcome	Name of the outcome column.
treatment	Name of the binary (0/1) treatment-group column.
post	Name of the binary (0/1) post-period column.
covariates	Optional character vector of covariate column names.
weights	Optional column of (sampling / survey) weights.
cluster	Optional cluster ID column for CR1 standard errors.
alpha	Significance level for confidence intervals (default 0.05).

Value

A list of class results; see morie_did_2x2.

`morie_did_sensitivity_analysis` *Sensitivity of DiD estimate to parallel-trends violations*

Description

For each δ, computes a bias-adjusted confidence set under the bound $|\text{bias}| \leq \delta\hat{\sigma}$ (Rambachan & Roth, 2023, conservative version).

Usage

```
morie_did_sensitivity_analysis(
  data,
  outcome,
  treatment,
  post,
  covariates = NULL,
  delta_range = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`covariates`	Optional character vector of covariate column names.
`delta_range`	Numeric vector of δ values to evaluate (default `seq(0, 2, 0.25)`).
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Details

For the full Rambachan-Roth fixed-length-confidence-interval (FLCI) procedure with event-time pre-trends prefer `HonestDiD::createSensitivityResults_relativeMagnitudes` on an event-study coefficient vector.

Value

A data frame with columns `delta`, `ci_lower`, `ci_upper`, `covers_zero`.

References

Rambachan, A., & Roth, J. (2023). A more credible approach to parallel trends. *Review of Economic Studies*, 90(5), 2555–2591.

`morie_did_staggered` *Staggered DiD via group-time ATTs with aggregation*

Description

Convenience wrapper around `morie_did_group_time_att` and `morie_did_aggregate_gt_att`. For the canonical CRAN aggregator interface see `did::aggte`.

Usage

```
morie_did_staggered(
  data,
  outcome,
  unit,
  time,
  treatment_time,
  covariates = NULL,
  n_bootstrap = 200L,
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	Panel data.
`outcome`	Outcome column.
`unit`	Unit identifier.
`time`	Calendar-time column (integer).
`treatment_time`	Column with treatment-onset period (use `Inf` for never-treated).
`covariates`	Optional covariates for doubly-robust estimation.
`n_bootstrap`	Number of bootstrap replications for inference (forwarded as `biters`).
`seed`	RNG seed (unused; retained for back-compat).
`alpha`	Significance level.

Value

A list with `group_time`, `overall`, `by_cohort`, `by_event_time`.

`morie_did_synthdid_estimate` *Synthetic DiD via* `synthdid::synthdid_estimate` *(explicit-name API)*

Description

Parallel to `morie_did_synthetic`, this is the explicit-name wrapper that surfaces the full `synthdid` estimator and its placebo / jackknife variance pieces. Use this when you want to pass through additional **synthdid** arguments or inspect the unit / time weights side-by-side; use `morie_did_synthetic` when you want the rmorie result-list shape consumed by `morie_did_*` downstream code.

Usage

```
morie_did_synthdid_estimate(
  panel,
  unit,
  time,
  treatment,
  outcome,
  vcov_method = "placebo",
  ...
)
```

Arguments

`panel`	Long-format balanced panel.
`unit`	Unit identifier column.
`time`	Time period column.
`treatment`	Binary treatment indicator that turns on at onset for treated units and is zero everywhere for controls (the **synthdid** W convention).
`outcome`	Outcome column.
`vcov_method`	Variance estimator passed to `stats::vcov.synthdid_estimate`: one of `"placebo"` (default), `"bootstrap"`, `"jackknife"`.
`...`	Additional arguments forwarded to `synthdid::synthdid_estimate` (e.g. `zeta`, `omega.intercept`).

Details

Wrapper-as-extender: rmorie already wraps **synthdid** once via `morie_did_synthetic`; this entry point gives MRM / paper callers the canonical Arkhangelsky et al. (2021) API with a `morie_*` name so they don't need to load **synthdid** directly.

Value

An S3 list of class `morie_did_synthdid_result` with elements `att`, `std_error`, `vcov_method`, `n_treated`, `n_control`, `n_pre`, `n_post`, `method`, and `raw` (the full `synthdid_estimate` object).

References

Arkhangelsky, D., Athey, S., Hirshberg, D. A., Imbens, G. W., & Wager, S. (2021). Synthetic difference-in-differences. *American Economic Review*, 111(12), 4088–4118.

See Also

`morie_did_synthetic`.

`morie_did_synthetic` *Synthetic Difference-in-Differences (Arkhangelsky et al., 2021)*

Description

Thin wrapper around `synthdid::synthdid_estimate`. Requires the **synthdid** package (`remotes::install_github("synth-inference/synthdid")`); synthdid is not on CRAN and has no comparable alternative.

Usage

```
morie_did_synthetic(
  data,
  outcome,
  unit,
  time,
  treatment_time,
  treated_units = NULL,
  zeta = NULL,
  n_bootstrap = 200L,
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	Balanced panel.
`outcome`, `unit`, `time`, `treatment_time`	Column names.
`treated_units`	Optional explicit list of treated unit IDs.
`zeta`	Optional regularisation parameter (auto-selected if NULL).
`n_bootstrap`	Bootstrap replications for placebo SE.
`seed`	RNG seed.
`alpha`	Significance level.

Value

A result list; see morie_did_2x2.

References

Arkhangelsky, D., et al. (2021). Synthetic difference-in-differences. *American Economic Review*, 111(12), 4088–4118.

---

morie_did_test_parallel_trends

Pre-trend test for parallel trends

Description

Regresses the outcome on group-by-time interactions in the pre-period and reports both per-period coefficients and a joint Wald (chi-square) test that they are all zero.

Usage

```
morie_did_test_parallel_trends(
  data,
  outcome,
  treatment,
  time,
  unit = NULL,
  cluster = NULL,
  pre_periods = NULL
)
```

Arguments

data	A data frame.
outcome	Outcome column name.
treatment	Binary treatment-group indicator.
time	Time column (integer-valued).
unit	Optional unit identifier (currently unused; reserved).
cluster	Cluster variable for robust SE.
pre_periods	Optional explicit list of pre-treatment times.

Details

For the Callaway-Sant'Anna pre-test on the group-time ATTs prefer did::conditional_did_pretest.

Value

A list with coefficients, joint_chi2 (and its alias joint_f_stat), joint_df, joint_p_value, parallel_trends_plausible.

`morie_did_triple_difference` *Triple-difference (DDD) estimator*

Description

Adds a third differencing dimension to the standard DiD specification.

Usage

```
morie_did_triple_difference(
  data,
  outcome,
  treatment,
  post,
  third_diff,
  covariates = NULL,
  cluster = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`third_diff`	Binary variable defining the additional differencing group.
`covariates`	Optional character vector of covariate column names.
`cluster`	Optional cluster ID column for CR1 standard errors.
`alpha`	Significance level for confidence intervals (default 0.05).

Value

A result list; see `morie_did_2x2`.

`morie_did_twoway_fe_weights` *Diagnose TWFE-DiD weights (de Chaisemartin & D'Haultfoeuille, 2020)*

Description

Thin interface to `TwoWayFEWeights::twowayfeweights`: returns the decomposition of the two-way fixed-effects DiD estimand into the weighted average of the $N \times T$ unit-time ATEs. Use this to quantify how many of the implicit comparisons receive negative weight, which is the canonical diagnostic for whether a TWFE specification can be interpreted as a convex combination of treatment effects.

Usage

```
morie_did_twoway_fe_weights(
  panel,
  group,
  time,
  treatment,
  outcome = NULL,
  type = "feTR",
  ...
)
```

Arguments

`panel`	A long-format balanced (or near-balanced) panel `data.frame`.
`group`	Name of the unit / group identifier column.
`time`	Name of the time period column.
`treatment`	Name of the binary or continuous treatment column.
`outcome`	Optional outcome column. When supplied, **TwoWayFEWeights** computes the weights AND the implied TWFE coefficient; when `NULL`, only the weights are returned (faster, dimension-free).
`type`	Weight type passed through to `TwoWayFEWeights::twowayfeweights`: `"feTR"` (default, feasible `TR` weights), `"feS"`, `"fdTR"`, or `"fdS"`. See the **TwoWayFEWeights** documentation.
`...`	Additional arguments forwarded to `TwoWayFEWeights::twowayfeweights`.

Details

Wrapper-as-extender: `morie_did_panel_fe` already estimates the TWFE coefficient; this function exposes the diagnostic side of the same backend so that downstream MRM analyses can flag heterogeneous- treatment-effects bias without leaving the rmorie API.

Value

An S3 list of class `morie_did_twfe_diagnostics` with elements `n_negative_weights`, `sum_weights`, `sum_negative_weights`, `share_negative_weights`, `method`, and `raw` (the full `twowayfeweights` object).

References

de Chaisemartin, C., & D'Haultfoeuille, X. (2020). Two-way fixed effects estimators with heterogeneous treatment effects. *American Economic Review*, 110(9), 2964–2996.

See Also

`morie_did_panel_fe`, `morie_did_chaisemartin_dhaultfoeuille`.

`morie_did_wild_cluster_bootstrap` *DiD with wild cluster bootstrap p-values (Cameron-Gelbach-Miller, 2008)*

Description

Recommended when the number of clusters is small (< 50). Uses a base-R Rademacher / Webb wild-cluster-bootstrap implementation. Earlier rmorie versions also delegated to `fwildclusterboot::boottest` when installed; that branch was dropped in 0.9.5.12 because fwildclusterboot is GitHub-only and transitively requires summclust, also GitHub-only, which made the CI dependency resolver unreliable. Callers who want **fwildclusterboot** should call it directly on a `feols` / `lm` fit.

Usage

```
morie_did_wild_cluster_bootstrap(
  data,
  outcome,
  treatment,
  post,
  cluster,
  covariates = NULL,
  n_bootstrap = 999L,
  weight_type = "rademacher",
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	A data frame containing the outcome, treatment, post and any covariate columns.
`outcome`	Name of the outcome column.
`treatment`	Name of the binary (0/1) treatment-group column.
`post`	Name of the binary (0/1) post-period column.
`cluster`	Optional cluster ID column for CR1 standard errors.
`covariates`	Optional character vector of covariate column names.
`n_bootstrap`	Number of bootstrap replications.
`weight_type`	`"rademacher"` (default) or `"webb"`.
`seed`	RNG seed.
`alpha`	Significance level for confidence intervals (default 0.05).

Value

A result list; see `morie_did_2x2`. `p_value` is the bootstrap p-value.

`morie_diffu_diffusion_forward`

DDPM forward (noising) process

Description

R parity for `morie.fn.diffu.diffusion_forward`.

Usage

```
morie_diffu_diffusion_forward(
  x0,
  t,
  betas = NULL,
  num_steps = 1000L,
  noise = NULL,
  seed = 0L
)
```

Arguments

`x0`	Clean sample.
`t`	Diffusion timestep (1..`num_steps`).
`betas`	Optional custom β schedule.
`num_steps`	Total diffusion steps (default 1000).
`noise`	Pre-generated Gaussian noise.
`seed`	RNG seed.

Details

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon$$

with linear β schedule from `1e-4` to `0.02`.

Value

Named list `(x_t, estimate, noise, alpha_bar, beta, method)`.

References

Ho, Jain & Abbeel (2020), NeurIPS.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

---

morie_download_bootstrap

Download bootstrap weight files from CKAN API

Description

Downloads large bootstrap weight CSVs that are too big to ship with the package. Data is cached in the user cache database for future use.

Usage

```
morie_download_bootstrap(
  survey = "all",
  limit = 32000L,
  db_path = NULL,
  con = NULL
)
```

Arguments

`survey`	One of `"csads_2021"`, `"csads_2023"`, `"csus_2019"`, `"csus_2023"`, or `"all"` (default).
`limit`	Max records per CKAN request (default 32000).
`db_path`	Optional path to a SQLite/DuckDB file (default backend).
`con`	Optional pre-opened DBI connection (overrides `db_path`).

Value

Invisibly, the number of CSV files successfully downloaded.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_dp_gaussian_mixture` *Bayesian nonparametric DP Gaussian mixture via* **dirichletprocess**

Description

Thin extender over `dirichletprocess::DirichletProcessGaussian` + `dirichletprocess::Fit` for a Bayesian nonparametric Dirichlet-process Gaussian mixture model (Ross & Markwick, 2018; MacEachern, 1994). Constructs the DP object on `y` and then runs the Gibbs sampler for `iterations` sweeps.

Usage

```
morie_dp_gaussian_mixture(y, iterations = 1000, ...)
```

Arguments

`y`	Numeric vector of observations to model with the DP Gaussian mixture.
`iterations`	Integer; number of Gibbs-sampler iterations to run via `dirichletprocess::Fit` (default `1000`).
`...`	Further arguments forwarded to `dirichletprocess::DirichletProcessGaussian` (e.g. `g0Priors`, `alphaPriors`, `mhDraws`, `verbose`).

Value

A list with `$method = "dirichletprocess::DirichletProcessGaussian + Fit"` and `$raw` (the fitted `dirichletprocess` object after the Gibbs run, containing the cluster assignments, cluster parameters, and concentration-parameter trace).

Examples

```
## Not run:
  if (requireNamespace("dirichletprocess", quietly = TRUE)) {
    set.seed(1)
    y <- c(stats::rnorm(50, -2), stats::rnorm(50, 2))
    morie_dp_gaussian_mixture(y, iterations = 200)
  }

## End(Not run)
```

`morie_dsp_acf_from_psd` *Autocorrelation from PSD (Wiener-Khinchin)*

Description

Inverse-rFFT of the PSD recovers the (biased) autocorrelation.

Usage

```
morie_dsp_acf_from_psd(psd)
```

Arguments

`psd`	PSD vector (one-sided).

Value

Numeric vector (autocorrelation).

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.3.

`morie_dsp_alpha_trimmed_mean` *Alpha-trimmed mean filter*

Description

Sliding-window mean after trimming `alpha` fraction of values from each tail of the sorted window. Robust to both Gaussian and impulsive noise; reduces to the mean filter when `alpha = 0` and to the median filter as `alpha -> 0.5`.

Usage

```
morie_dsp_alpha_trimmed_mean(x, window = 5L, alpha = 0.2)
```

Arguments

`x`	Numeric vector.
`window`	Window length. Default 5.
`alpha`	Trim fraction (0 <= alpha < 0.5). Default 0.2.

Value

Filtered vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.4.

`morie_dsp_amplitude_histogram` *Amplitude histogram features*

Description

Equal-width histogram of `x` with `n_bins` bins. Returns counts, bin centres, probabilities, and edges (parallel to numpy.histogram).

Usage

```
morie_dsp_amplitude_histogram(x, n_bins = 50L)
```

Arguments

`x`	Numeric vector.
`n_bins`	Number of bins. Default 50.

Value

List with `counts`, `centers`, `probabilities`, `edges`.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.6.

`morie_dsp_arc_length` *Signal arc length*

Description

Polygonal length: `sum(sqrt(1 + diff(x)^2))`. Curve length under unit time-step.

Usage

```
morie_dsp_arc_length(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.3.

`morie_dsp_band_power` *Bandpower over* `[f_low, f_high]`

Description

Trapezoid-equivalent rectangular integration of PSD over a band.

Usage

```
morie_dsp_band_power(psd, freqs, f_low, f_high)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.
`f_low`	Lower edge (Hz).
`f_high`	Upper edge (Hz).

Value

Band power (units of PSD x Hz).

References

Rangayyan & Krishnan (2015), Ch. 6.

`morie_dsp_baseline_correlation`
Baseline-corrected Pearson correlation

Description

`cor(x - mean(x), y - mean(y))` with explicit zero-norm guard.

Usage

```
morie_dsp_baseline_correlation(x, y)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.

Value

Scalar in `[-1, 1]`.

References

Rangayyan & Krishnan (2015), Ch. 5.

`morie_dsp_centroidal_time` *Centroidal time*

Description

Time-domain centre of energy: `sum(t * x^2) / sum(x^2)`.

Usage

```
morie_dsp_centroidal_time(x, fs = 1)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.

Value

Scalar (seconds).

References

Rangayyan & Krishnan (2015), Ch. 5.

`morie_dsp_coherence` *Coherence-squared spectrum*

Description

Delegates to `signal::coherence` when present; otherwise builds a Welch-style estimator from `morie_dsp_psd_welch` and a parallel CSD.

Usage

```
morie_dsp_coherence(x, y, fs = 1, nperseg = 256L)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.
`nperseg`	Segment length. Default 256.

Value

List with `freqs` and `coh` (magnitude squared coherence).

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.9.

`morie_dsp_coherence_spectrum` *Coherence spectrum (alias to morie_dsp_coherence)*

Description

Convenience pass-through so detection-side callers can stay within the `dsp_detection` namespace.

Usage

```
morie_dsp_coherence_spectrum(x, y, fs = 1, nperseg = 256L)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.
`nperseg`	Segment length. Default 256.

Value

Same as `morie_dsp_coherence`.

References

Rangayyan & Krishnan (2015), Ch. 4 & Ch. 6.

morie_dsp_comb *Comb filter built from cascaded notches*

Description

Successively notches the fundamental and its first `n_harmonics` harmonics. Each notch below Nyquist is applied; aliased harmonics are skipped.

Usage

```
morie_dsp_comb(x, fundamental, fs, n_harmonics = 5L, q = 30)
```

Arguments

`x`	Numeric vector.
`fundamental`	Fundamental frequency (Hz), e.g. 60 for North American mains.
`fs`	Sampling frequency (Hz).
`n_harmonics`	Number of harmonics to cancel. Default 5.
`q`	Quality factor per notch. Default 30.

Value

Filtered vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.7.

morie_dsp_complex_cepstrum *Complex cepstrum*

Description

`IFFT(log|X| + j * unwrap(angle(X)))`. Returns cepstrum and quefrency indices.

Usage

```
morie_dsp_complex_cepstrum(x)
```

Arguments

`x`	Numeric vector.

Value

List with `cepstrum` and `quefrency`.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.10; Oppenheim & Schafer (2010).

`morie_dsp_complex_demodulation`
Complex demodulation around a carrier

Description

Multiplies `x` by `exp(-j 2 pi fc t)`, low-passes via Butterworth (requires `signal`), and returns envelope and unwrapped phase.

Usage

```
morie_dsp_complex_demodulation(x, fc, fs = 1)
```

Arguments

`x`	Numeric vector.
`fc`	Carrier frequency (Hz).
`fs`	Sampling frequency (Hz). Default 1.

Value

List with `envelope` and `phase`, both length(x).

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.8.

`morie_dsp_crest_factor`
Crest factor (peak / RMS)

Description

`sqrt(2)` for a pure sine; large values indicate spiky waveforms.

Usage

```
morie_dsp_crest_factor(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.2.

`morie_dsp_cross_correlation`
Normalised cross-correlation up to a maximum lag

Description

Centres both inputs (subtracts the mean) and divides by the geometric mean of their L2 norms. Returns lags `-max_lag .. +max_lag`.

Usage

```
morie_dsp_cross_correlation(x, y, max_lag = NULL)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector, same length as `x`.
`max_lag`	Maximum lag (defaults to `length(x) - 1`).

Value

Numeric vector of length `2 * max_lag + 1`.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.4.

`morie_dsp_csd` *Cross-spectral density (Welch)*

Description

Hamming-windowed averaged CSD. Delegates to `signal::cpsd` if present; otherwise computed from the same FFT loop as the coherence fallback.

Usage

```
morie_dsp_csd(x, y, fs = 1, nperseg = 256L)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.
`nperseg`	Segment length. Default 256.

Value

List with `freqs` (Hz) and `csd` (complex).

References

Rangayyan & Krishnan (2015), Ch. 4 & Ch. 6.

`morie_dsp_cv` *Coefficient of variation*

Description

`sd(x) / |mean(x)|`. `Inf` when the mean is zero.

Usage

```
morie_dsp_cv(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 3.

`morie_dsp_derivative_detect` *Derivative-based peak detection*

Description

Locates samples where the first derivative crosses zero from positive to non-positive AND the prior slope magnitude exceeds `threshold_factor * max(|dx|)`.

Usage

```
morie_dsp_derivative_detect(x, fs = 1, threshold_factor = 0.5)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.
`threshold_factor`	Slope threshold fraction. Default 0.5.

Value

Integer vector of peak indices.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.3.

`morie_dsp_dicrotic_notch`
Dicrotic-notch detection in pulse waves

Description

Finds prominent minima in the second derivative outside the systolic onset window. Requires `signal::findpeaks`.

Usage

```
morie_dsp_dicrotic_notch(pulse, fs = 125)
```

Arguments

`pulse`	Numeric pulse vector.
`fs`	Sampling frequency (Hz). Default 125.

Value

Integer vector of notch indices.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.8.

`morie_dsp_ensemble_average`
Ensemble average over fixed-length segments

Description

Averages a `(n_segments, n_samples)` matrix down its rows. Standard synchronous-averaging recipe for repeated stimulus responses (e.g. evoked potentials).

Usage

```
morie_dsp_ensemble_average(segments)
```

Arguments

`segments`	Numeric matrix (rows = trials, cols = samples).

Value

Row mean as a length-`ncol` vector.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.3.

---

morie_dsp_entropy_histogram

Shannon entropy from amplitude histogram

Description

`-sum(p log2 p)` over the histogram probabilities.

Usage

```
morie_dsp_entropy_histogram(x, n_bins = 50L)
```

Arguments

`x`	Numeric vector.
`n_bins`	Number of bins. Default 50.

Value

Scalar (bits).

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.6.

---

morie_dsp_even_odd

Even-odd decomposition of a finite signal

Description

Splits `x` into its even (symmetric) and odd (anti-symmetric) components about the centre: `x_even = (x + rev(x))/2`, `x_odd = (x - rev(x))/2`.

Usage

```
morie_dsp_even_odd(x)
```

Arguments

`x`	Numeric vector.

Value

List with `even` and `odd`.

References

Rangayyan & Krishnan (2015), Ch. 3.

`morie_dsp_fbm_synthesis`
Fractional Brownian motion synthesis (1/f^beta)

Description

Generates a length-`N` fBm via spectral shaping of white noise with `|f|^{-(H + 0.5)}`, then cumulative integration. `H = 0.5` recovers ordinary Brownian motion.

Usage

```
morie_dsp_fbm_synthesis(N, H = 0.5)
```

Arguments

`N`	Length.
`H`	Hurst exponent in (0, 1). Default 0.5.

Value

Numeric vector, length `N`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.8; Mandelbrot & Van Ness (1968).

`morie_dsp_form_factor`
Form factor (RMS / mean-absolute)

Description

`1.11` for a pure sine; deviations diagnose waveshape changes.

Usage

```
morie_dsp_form_factor(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.2.

`morie_dsp_fractal_dim_psd`
Fractal dimension from log-log PSD slope

Description

Fits `log10(psd) ~ log10(f)` on positive bins; with slope `-beta`, returns the 1/f fractal dimension `(5 - beta) / 2`. Falls back to 1.5 (Brownian) when fewer than two valid bins exist.

Usage

```
morie_dsp_fractal_dim_psd(psd, freqs)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.

Value

Scalar fractal dimension.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.8; Eke et al. (2002).

`morie_dsp_hann_filter`
Hann-windowed smoothing filter

Description

Convolves `x` with a normalised Hann (raised-cosine) window of length `window`. Less ringing than the boxcar.

Usage

```
morie_dsp_hann_filter(x, window = 5L)
```

Arguments

`x`	Numeric vector.
`window`	Window length. Default 5.

Value

Smoothed vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3.

`morie_dsp_higuchi_fd` *Higuchi fractal dimension*

Description

Slope of `log(L(k))` vs. `log(1/k)` over `k = 1..kmax` curve-length scales. Returns a value in approximately `[1, 2]` for real signals.

Usage

```
morie_dsp_higuchi_fd(x, kmax = 10L)
```

Arguments

`x`	Numeric vector.
`kmax`	Maximum scale. Default 10.

Value

Scalar fractal dimension.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.7; Higuchi (1988).

`morie_dsp_hilbert_envelope` *Hilbert envelope*

Description

Analytic-signal magnitude via the Hilbert transform. Delegates to `signal::hilbert` when available.

Usage

```
morie_dsp_hilbert_envelope(x)
```

Arguments

`x`	Numeric vector.

Value

Numeric vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.6.

`morie_dsp_hjorth` *All three Hjorth parameters*

Description

All three Hjorth parameters

Usage

```
morie_dsp_hjorth(x)
```

Arguments

`x`	Numeric vector.

Value

Named list: `activity`, `mobility`, `complexity`.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.5; Hjorth (1970).

`morie_dsp_hjorth_activity`
Hjorth activity (variance)

Description

First Hjorth parameter: signal variance.

Usage

```
morie_dsp_hjorth_activity(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.5; Hjorth (1970).

`morie_dsp_hjorth_complexity`
Hjorth complexity

Description

`mobility(diff(x)) / mobility(x)`; bandwidth-like descriptor (1 for a pure sinusoid).

Usage

```
morie_dsp_hjorth_complexity(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.5.

`morie_dsp_hjorth_mobility` *Hjorth mobility*

Description

`sqrt(var(diff(x)) / var(x))`; proportional to mean frequency.

Usage

```
morie_dsp_hjorth_mobility(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.5.

`morie_dsp_homomorphic` *Homomorphic high-pass filter*

Description

Log -> rFFT high-pass at `cutoff` Hz -> exp. Reduces multiplicative baseline drift in non-negative envelopes.

Usage

```
morie_dsp_homomorphic(x, cutoff = 0.1, fs = 1)
```

Arguments

`x`	Numeric vector.
`cutoff`	Cutoff frequency (Hz). Default 0.1.
`fs`	Sampling frequency (Hz). Default 1.

Value

Numeric vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.9; Oppenheim & Schafer (2010).

---

morie_dsp_hr_from_rr *Heart rate from RR intervals (BPM)*

Description

`60 / rr`, with zero-RR turned into `NA`.

Usage

```
morie_dsp_hr_from_rr(rr_intervals)
```

Arguments

`rr_intervals`	Numeric vector (seconds).

Value

Numeric vector of BPM values.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.8.

---

morie_dsp_integrated_emg *Integrated EMG (sum of absolute values)*

Description

Integrated EMG (sum of absolute values)

Usage

```
morie_dsp_integrated_emg(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.3.

morie_dsp_katz_fd *Katz / box-counting fractal dimension*

Description

Coarse box-counting on the amplitude-normalised signal across `n_scales` log-spaced box widths; returns the slope of `log N(s)` vs. `log(1/s)`.

Usage

```
morie_dsp_katz_fd(x, n_scales = 10L)
```

Arguments

`x`	Numeric vector.
`n_scales`	Number of box sizes. Default 10.

Value

Scalar fractal dimension.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.7; Katz (1988).

morie_dsp_lms *LMS adaptive filter (Widrow-Hoff)*

Description

Least-mean-squares adaptive transversal filter. Returns the filter output `y` and instantaneous error `e = d - y`. Coefficient update: `w <- w + 2 * mu * e[i] * x_seg`.

Usage

```
morie_dsp_lms(x, d, order = 16L, mu = 0.01)
```

Arguments

`x`	Input (reference) vector.
`d`	Desired vector, same length as `x`.
`order`	Filter order (taps). Default 16.
`mu`	Step size. Default 0.01.

Value

List with elements `y` and `e`, both length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.6; Widrow & Stearns (1985).

`morie_dsp_matched` *Matched filter (time-reversed template correlator)*

Description

Optimal linear filter against white Gaussian noise: convolves `x` with the time-reversed template, normalised by `||template||`.

Usage

```
morie_dsp_matched(x, template)
```

Arguments

`x`	Numeric vector.
`template`	Reference waveform.

Value

Matched-filter output, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.5.

`morie_dsp_mean_abs` *Mean absolute value*

Description

Mean absolute value

Usage

```
morie_dsp_mean_abs(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.3.

`morie_dsp_mean_frequency`
Mean frequency from PSD

Description

`m1 / m0`; equals the first moment normalised by total power.

Usage

```
morie_dsp_mean_frequency(psd, freqs)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.

Value

Scalar mean frequency (Hz).

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.6.

`morie_dsp_median_filter`
Median filter

Description

Sliding-window median. Robust to impulsive (salt-and-pepper) noise.

Usage

```
morie_dsp_median_filter(x, kernel_size = 5L)
```

Arguments

`x`	Numeric vector.
`kernel_size`	Odd positive integer kernel length. Default 5.

Value

Filtered vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.4.

`morie_dsp_median_frequency`

Median frequency from PSD

Description

Frequency at which the cumulative spectrum reaches half the total power. Robust to high-frequency outliers vs. the mean frequency.

Usage

```
morie_dsp_median_frequency(psd, freqs)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.

Value

Scalar (Hz).

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.6.

`morie_dsp_min_phase` *Minimum-phase correspondent via cepstral folding*

Description

Folds the real cepstrum to the causal half then re-exponentiates to produce a minimum-phase sequence with the same magnitude spectrum as `x` (approximate).

Usage

```
morie_dsp_min_phase(x)
```

Arguments

`x`	Numeric vector.

Value

Numeric vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 5; Oppenheim & Schafer (2010).

`morie_dsp_moving_average`
Moving-average filter (boxcar)

Description

Length-`window` boxcar convolution in "same" mode (output length equals input length, edges biased by zero-padding).

Usage

```
morie_dsp_moving_average(x, window = 5L)
```

Arguments

`x`	Numeric vector.
`window`	Positive integer kernel length. Default 5.

Value

Numeric vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.3.

morie_dsp_myopulse_rate *Myopulse percentage rate*

Description

Fraction of samples with `|x| > threshold` (defaults to `2 * sd(x)`).

Usage

```
morie_dsp_myopulse_rate(x, threshold = NULL)
```

Arguments

`x`	Numeric vector.
`threshold`	Optional threshold.

Value

Scalar in `[0, 1]`.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.4.

morie_dsp_nlms *NLMS adaptive filter*

Description

Normalised LMS: divides the step by the instantaneous input power `x_seg' x_seg + eps`, giving robust convergence over a wider range of input scales.

Usage

```
morie_dsp_nlms(x, d, order = 16L, mu = 0.5, eps = 1e-08)
```

Arguments

`x`	Input (reference) vector.
`d`	Desired vector, same length as `x`.
`order`	Filter order (taps). Default 16.
`mu`	Normalised step size (typically $0 < \mu < 2$). Default 0.5.
`eps`	Power-floor for division. Default 1e-8.

Value

List with `y`, `e`.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.6.

`morie_dsp_notch` *IIR notch filter (single frequency)*

Description

Wraps `signal::butter` style IIR notch via the `signal` package's `iirnotch` / `filtfilt`. Falls back to a stop with `NotYetPorted` if `signal` is unavailable.

Usage

```
morie_dsp_notch(x, freq, fs, q = 30)
```

Arguments

`x`	Numeric vector.
`freq`	Notch centre frequency (Hz).
`fs`	Sampling frequency (Hz).
`q`	Quality factor. Default 30.

Value

Filtered vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.7.

`morie_dsp_onset_detect`
Energy-onset detection

Description

Smooths `x^2` with a `energy_window_ms` boxcar and flags samples where the smoothed energy crosses `threshold_factor * median(energy)`. Hysteresis: returns to "off" only when energy drops below baseline.

Usage

```
morie_dsp_onset_detect(x, fs, energy_window_ms = 20, threshold_factor = 3)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz).
`energy_window_ms`	Smoothing window (ms). Default 20.
`threshold_factor`	Multiplier on baseline median. Default 3.

Value

Integer vector of onset indices.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.5.

`morie_dsp_pan_tompkins`
Pan-Tompkins QRS detector

Description

Bandpass (5-15 Hz) -> differentiate -> square -> moving-window integrate -> adaptive threshold with refractory period. Requires `signal` for the Butterworth bandpass.

Usage

```
morie_dsp_pan_tompkins(ecg, fs = 360)
```

Arguments

`ecg`	ECG vector.
`fs`	Sampling frequency (Hz). Default 360.

Value

Integer vector of QRS sample indices.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.7; Pan & Tompkins (1985).

`morie_dsp_parzen_pdf` *Parzen kernel density estimate*

Description

Gaussian-kernel KDE on a uniform grid. Bandwidth defaults to Silverman's `1.06 * sd(x) * n^{-1/5}`.

Usage

```
morie_dsp_parzen_pdf(x, bandwidth = NULL, n_points = 100L)
```

Arguments

`x`	Numeric vector.
`bandwidth`	Optional kernel bandwidth.
`n_points`	Grid size. Default 100.

Value

List with `grid` and `density`.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.6; Parzen (1962); Silverman (1986).

`morie_dsp_psd_bartlett`
Bartlett PSD estimate

Description

Splits `x` into `n_segments` non-overlapping equal segments, periodograms each, and averages the result. Variance scales as `1 / n_segments` at the cost of frequency resolution.

Usage

```
morie_dsp_psd_bartlett(x, fs = 1, n_segments = 8L)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.
`n_segments`	Number of equal segments. Default 8.

Value

List with `freqs` and `psd`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.4; Bartlett (1948).

`morie_dsp_psd_periodogram`
Periodogram PSD estimate

Description

One-sided rFFT-based periodogram. Inner bins are doubled to fold negative-frequency power into the one-sided spectrum.

Usage

```
morie_dsp_psd_periodogram(x, fs = 1)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.

Value

List with `freqs` and `psd`, both length `floor(N/2)+1`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.4.

`morie_dsp_psd_to_db` *Convert PSD to decibels*

Description

`10 * log10(max(psd, 1e-20))`.

Usage

```
morie_dsp_psd_to_db(psd)
```

Arguments

`psd`	PSD vector.

Value

PSD in dB.

References

Rangayyan & Krishnan (2015), Ch. 6.

`morie_dsp_psd_welch` *Welch PSD estimate (delegated to signal::pwelch / specgram)*

Description

Thin wrapper that prefers the `signal` package's `pwelch`-style routine for the production estimator; falls back to a pure-R Hamming-windowed averaged periodogram when `signal` is unavailable.

Usage

```
morie_dsp_psd_welch(x, fs = 1, nperseg = 256L, noverlap = NULL)
```

Arguments

`x`	Numeric vector.
`fs`	Sampling frequency (Hz). Default 1.
`nperseg`	Segment length. Default 256.
`noverlap`	Overlap in samples. Default `nperseg %/% 2`.

Value

List with `freqs` and `psd`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.4; Welch (1967).

`morie_dsp_qrs_features` *QRS-style waveform descriptors*

Description

Returns peak amplitude, duration (samples), absolute area (trapezoid), up/down slopes, and peak index for a single beat.

Usage

```
morie_dsp_qrs_features(beat)
```

Arguments

`beat`	Numeric vector covering one beat.

Value

Named list of features.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.9.

morie_dsp_rls *RLS adaptive filter*

Description

Recursive least-squares with forgetting factor `lam` and initial inverse-correlation `P0 = delta * I`. Faster convergence than LMS at the cost of O(order^2) per sample.

Usage

```
morie_dsp_rls(x, d, order = 16L, lam = 0.99, delta = 100)
```

Arguments

`x`	Input (reference) vector.
`d`	Desired vector, same length as `x`.
`order`	Filter order (taps). Default 16.
`lam`	Forgetting factor in (0, 1]. Default 0.99.
`delta`	Initial P diagonal. Default 100.

Value

List with `y`, `e`.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.6; Haykin (2002).

morie_dsp_rms *Root mean square*

Description

`sqrt(mean(x^2))`.

Usage

```
morie_dsp_rms(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar RMS.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.2.

`morie_dsp_ruler_fd` *Ruler / divider fractal dimension*

Description

Slope-based estimator from polygonal lengths at log-spaced ruler sizes. Equivalent up to sign convention to Higuchi.

Usage

```
morie_dsp_ruler_fd(x, n_rulers = 10L)
```

Arguments

`x`	Numeric vector.
`n_rulers`	Number of ruler sizes. Default 10.

Value

Scalar fractal dimension.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.7.

`morie_dsp_shannon_energy` *Shannon-energy envelope*

Description

`-x^2 * log(x^2)` with a small floor to guard log(0). Amplifies moderate-energy components, useful as a preprocessor for heart-sound segmentation.

Usage

```
morie_dsp_shannon_energy(x)
```

Arguments

`x`	Numeric vector.

Value

Numeric vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.5; Liang et al. (1997).

`morie_dsp_shape_factor`
Shape factor (mean-absolute / mean-sqrt-absolute-squared)

Description

Dimensionless waveshape descriptor: `E|x| / (E sqrt|x|)^2`.

Usage

```
morie_dsp_shape_factor(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.2.

`morie_dsp_slope_sign_changes`
Slope sign changes

Description

Sign changes in `diff(x)` where the absolute next slope exceeds `threshold`. Hudgins TD feature.

Usage

```
morie_dsp_slope_sign_changes(x, threshold = 0)
```

Arguments

`x`	Numeric vector.
`threshold`	Magnitude threshold. Default 0.

Value

Integer count.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.4.

`morie_dsp_snr` *Signal-to-noise ratio estimate (dB)*

Description

`10 * log10(mean(signal^2) / mean(noise^2))`. Returns `Inf` when noise power is zero.

Usage

```
morie_dsp_snr(signal, noise)
```

Arguments

`signal`	Numeric vector.
`noise`	Numeric vector.

Value

SNR in dB.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.2.

`morie_dsp_snr_improvement`
SNR improvement attributable to a filter (dB)

Description

Compares SNR of (clean vs. clean-noisy) before filtering with SNR of (clean vs. clean-filtered) after; positive values mean the filter reduced noise relative to clean.

Usage

```
morie_dsp_snr_improvement(x_noisy, x_clean, x_filtered)
```

Arguments

`x_noisy`	Observed noisy vector.
`x_clean`	Clean reference.
`x_filtered`	Filter output.

Value

Delta SNR in dB.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.2.

`morie_dsp_spectral_edge`
Spectral edge frequency (e.g. SEF95)

Description

Frequency below which `pct` fraction of total power lies. SEF95 (`pct = 0.95`) is a classical EEG depth-of-anaesthesia marker.

Usage

```
morie_dsp_spectral_edge(psd, freqs, pct = 0.95)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.
`pct`	Cumulative fraction in (0, 1]. Default 0.95.

Value

Scalar (Hz).

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.6.

`morie_dsp_spectral_entropy`
Spectral entropy (Shannon, base 2)

Description

Normalises PSD to a probability mass function and returns its Shannon entropy in bits.

Usage

```
morie_dsp_spectral_entropy(psd)
```

Arguments

`psd`	PSD vector.

Value

Scalar in `[0, log2(length(psd))]`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.7; Inouye et al. (1991).

`morie_dsp_spectral_flatness`
Spectral flatness (Wiener entropy)

Description

Geometric-to-arithmetic mean ratio of the positive PSD bins; values near 1 indicate a white spectrum, near 0 indicate tonal concentration.

Usage

```
morie_dsp_spectral_flatness(psd)
```

Arguments

`psd`	PSD vector.

Value

Scalar in `[0, 1]`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.7.

`morie_dsp_spectral_kurtosis`
Spectral kurtosis from PSD

Description

Standardised fourth central moment of frequency under the PSD treated as a probability mass.

Usage

```
morie_dsp_spectral_kurtosis(psd, freqs)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 6.

`morie_dsp_spectral_moment`
k-th spectral moment

Description

`m_k = sum(freqs^k * psd) * df`. Used to derive mean, median, edge, and higher-order frequency descriptors.

Usage

```
morie_dsp_spectral_moment(psd, freqs, order = 0L)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.
`order`	Moment order. Default 0.

Value

Scalar moment.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.6.

`morie_dsp_spectral_ratio`

Spectral power ratio between two bands

Description

Returns `bandpower(psd, band1) / bandpower(psd, band2)`.

Usage

```
morie_dsp_spectral_ratio(psd, freqs, band1, band2)
```

Arguments

`psd`	PSD vector.
`freqs`	Matching frequency vector.
`band1`	Length-2 numeric (low, high) Hz.
`band2`	Length-2 numeric (low, high) Hz.

Value

Scalar ratio.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.6.

`morie_dsp_synchronized_average` *Synchronized average around trigger indices*

Description

Extracts length-`window` epochs centred on each `trigger_indices` value and averages them. Out-of-bounds triggers are dropped.

Usage

```
morie_dsp_synchronized_average(x, trigger_indices, window = 100L)
```

Arguments

`x`	Numeric vector.
`trigger_indices`	Integer vector of 1-based event indices.
`window`	Epoch length (centred). Default 100.

Value

Mean epoch, length `window`.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.3.

`morie_dsp_t_wave` *T-wave detection by post-QRS argmax search*

Description

For each QRS index, searches `[loc + 0.2 * fs, loc + 0.5 * fs]` for the absolute maximum and records its global index.

Usage

```
morie_dsp_t_wave(ecg, qrs_locs, fs = 360)
```

Arguments

`ecg`	ECG vector.
`qrs_locs`	QRS indices (1-based) from a detector like `morie_dsp_pan_tompkins`.
`fs`	Sampling frequency (Hz). Default 360.

Value

Integer vector of T-peak indices.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.8.

`morie_dsp_teager_energy`
Teager-Kaiser energy operator

Description

`psi[n] = x[n]^2 - x[n-1] * x[n+1]`; sensitive to instantaneous amplitude AND frequency.

Usage

```
morie_dsp_teager_energy(x)
```

Arguments

`x`	Numeric vector.

Value

Numeric vector, length(x); ends are zero.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.5; Kaiser (1990).

`morie_dsp_template_match`
Normalised template matching

Description

Slides `template` across `x`, computing the centred Pearson-style correlation per offset. Returns indices and correlation values meeting `threshold`.

Usage

```
morie_dsp_template_match(x, template, threshold = 0.7)
```

Arguments

`x`	Numeric vector.
`template`	Numeric vector (shorter than `x`).
`threshold`	Minimum correlation. Default 0.7.

Value

List with `indices` (1-based) and `correlations`.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.4.

`morie_dsp_threshold_detect`

Threshold-based event detection

Description

Returns sample indices (1-based) where `x` crosses `threshold` in the chosen direction. `min_distance` enforces a minimum gap (in samples) between successive events.

Usage

```
morie_dsp_threshold_detect(
  x,
  threshold,
  min_distance = 1L,
  direction = "above"
)
```

Arguments

`x`	Numeric vector.
`threshold`	Scalar threshold.
`min_distance`	Minimum gap between events. Default 1.
`direction`	One of "above", "below", "either". Default "above".

Value

Integer vector of detected indices.

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.2.

`morie_dsp_turning_points`
Turning-points stationarity test

Description

Counts strict turning points and z-scores against the i.i.d. expectation `2(n-2)/3`. `|z| < 1.96` is consistent with weak stationarity at the 5 percent level.

Usage

```
morie_dsp_turning_points(x)
```

Arguments

`x`	Numeric vector.

Value

List with `turning_points`, `expected`, `z_statistic`, `stationary`.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.2.

`morie_dsp_turns_count`
Willison turns count

Description

Counts adjacent sign changes in `diff(x)` whose absolute slope difference exceeds `threshold`. Used as a fatigue/load proxy in sEMG analysis.

Usage

```
morie_dsp_turns_count(x, threshold = 0)
```

Arguments

`x`	Numeric vector.
`threshold`	Slope-difference threshold. Default 0.

Value

Integer count.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.4; Willison (1964).

`morie_dsp_variance_ratio`
Variance ratio (x vs. y)

Description

`var(x) / var(y)`; `Inf` if `var(y) == 0`.

Usage

```
morie_dsp_variance_ratio(x, y)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5.

`morie_dsp_waveform_length`
Waveform length (total variation)

Description

Sum of absolute first differences. Standard sEMG descriptor.

Usage

```
morie_dsp_waveform_length(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.3; Hudgins et al. (1993).

`morie_dsp_waveform_length_norm`
Per-sample (normalised) waveform length

Description

`morie_dsp_waveform_length(x) / length(x)`.

Usage

```
morie_dsp_waveform_length_norm(x)
```

Arguments

`x`	Numeric vector.

Value

Scalar.

References

Rangayyan & Krishnan (2015), Ch. 5.

`morie_dsp_wiener_filter`
Wiener filter (frequency domain)

Description

Classical scalar Wiener gain in the rFFT domain: `H(f) = Pxx(f) / (Pxx(f) + Pnn(f))`. With `noise_psd = NULL` the noise PSD is assumed flat at `noise_fraction * mean(Pxx)`.

Usage

```
morie_dsp_wiener_filter(x, noise_psd = NULL, noise_fraction = 0.1)
```

Arguments

`x`	Numeric vector (signal + noise).
`noise_psd`	Optional noise PSD, length `floor(N/2)+1`.
`noise_fraction`	Fallback flat-noise scale. Default 0.1.

Value

Filtered vector, length(x).

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.5.

`morie_dsp_wiener_hopf`
Solve the Wiener-Hopf normal equations

Description

Returns `w = solve(Rxx, rxd)`. Used as the optimal-FIR Wiener solution; equivalent to `lm.fit` on Toeplitz inputs.

Usage

```
morie_dsp_wiener_hopf(Rxx, rxd)
```

Arguments

`Rxx`	Symmetric autocorrelation matrix (order x order).
`rxd`	Cross-correlation vector (length order).

Value

Optimal tap-weight vector.

References

Rangayyan & Krishnan (2015), Ch. 3, sec. 3.5.

`morie_dsp_willison_amplitude`
Willison amplitude

Description

Count of `|diff(x)| > threshold` (defaults to `sd(x)`).

Usage

```
morie_dsp_willison_amplitude(x, threshold = NULL)
```

Arguments

`x`	Numeric vector.
`threshold`	Optional threshold (default `sd(x)`).

Value

Integer count.

References

Rangayyan & Krishnan (2015), Ch. 5, sec. 5.4.

`morie_dsp_window` *Window function generator*

Description

Returns a length-`N` window vector of the requested type. Supports hamming, hann/hanning, blackman, bartlett/triangular, kaiser (beta = 14), and rectangular/boxcar. Unknown types default to hamming.

Usage

```
morie_dsp_window(N, wtype = "hamming")
```

Arguments

`N`	Window length.
`wtype`	Type string. Default "hamming".

Value

Numeric vector of length `N`.

References

Rangayyan & Krishnan (2015), Ch. 6, sec. 6.5.

morie_dsp_zero_crossing *Zero-crossing rate*

Description

Whole-signal ZCR if `frame_length = NULL`; otherwise per-frame ZCR over consecutive non-overlapping frames.

Usage

```
morie_dsp_zero_crossing(x, frame_length = NULL)
```

Arguments

`x`	Numeric vector.
`frame_length`	Optional frame length.

Value

Scalar (whole signal) or numeric vector (per frame).

References

Rangayyan & Krishnan (2015), Ch. 4, sec. 4.3.

morie_e_value *Compute E-value for unmeasured confounding*

Description

The E-value quantifies the minimum strength of confounding association needed to fully explain away an observed treatment effect:

$$E = RR + \sqrt{RR \cdot (RR - 1)}$$

Usage

```
morie_e_value(rr, rr_lower = NULL)
```

Arguments

`rr`	Risk ratio estimate (> 0). Supply > 1; if < 1, pass its reciprocal.
`rr_lower`	Lower bound of the 95\ E-value for CI).

Details

For a risk ratio $RR < 1$, use $1/RR$ before applying the formula.

Thin wrapper over `EValue::evalue()` when **EValue** is installed; falls back to the inline closed-form computation otherwise. Both arms produce numerically identical answers (the formula above is the EValue closed-form for RR estimands).

Value

Named list: `morie_e_value`, `e_value_ci` (for the CI bound).

References

VanderWeele TJ, Ding P (2017). Sensitivity analysis in observational research: introducing the E-value. *Annals of Internal Medicine*, 167(4):268-274.

Examples

```
morie_e_value(rr = 3.9, rr_lower = 2.4)
```

`morie_effects_comparisons`

Contrasts / comparisons via **marginaleffects**

Description

Thin extender over `marginaleffects::comparisons()` for unit- level treatment-effect contrasts (counterfactual differences, ratios, etc.).

Usage

```
morie_effects_comparisons(model, variables = NULL, ...)
```

Arguments

`model`	A fitted model object.
`variables`	Character vector or named list of variables to contrast (see `marginaleffects::comparisons()`). When `NULL`, marginaleffects' default (all model variables) is used.
`...`	Further arguments forwarded to `marginaleffects::comparisons()`.

Value

A `marginaleffects` data frame.

`morie_effects_emmeans`

Estimated marginal means via **emmeans**

Description

Thin extender over `emmeans::emmeans()`. The fitted model is passed through unchanged; `specs` follows the usual emmeans formula / list interface. Use `emmeans::pairs()` or `emmeans::contrast()` on the returned object for pairwise or custom contrasts.

Usage

```
morie_effects_emmeans(model, specs, ...)
```

Arguments

`model`	A fitted model object (`lm`, `glm`, `lmerMod`, ...).
`specs`	Specification for the marginal means – a formula (e.g. `~ treatment`), character vector of factor names, or a list, exactly as accepted by `emmeans::emmeans()`.
`...`	Further arguments forwarded to `emmeans::emmeans()`.

Value

An `emmGrid` object.

`morie_effects_predictions`

Adjusted predictions via **marginaleffects**

Description

Thin extender over `marginaleffects::predictions()` for unit- level or grid-level adjusted predictions.

Usage

```
morie_effects_predictions(model, newdata = NULL, ...)
```

Arguments

`model`	A fitted model object supported by **insight** / **marginaleffects**.
`newdata`	Optional data frame for which to predict. Defaults to the model frame when `NULL` (the marginaleffects default).
`...`	Further arguments forwarded to `marginaleffects::predictions()`.

Value

A `marginaleffects` data frame.

`morie_effects_slopes` *Marginal slopes (partial derivatives) via* **marginaleffects**

Description

Thin extender over `marginaleffects::slopes()` for continuous marginal effects (Stata-style `margins, dydx()`).

Usage

```
morie_effects_slopes(model, variables = NULL, ...)
```

Arguments

`model`	A fitted model object.
`variables`	Character vector of focal variables. When `NULL`, the marginaleffects default (all continuous predictors) is used.
`...`	Further arguments forwarded to `marginaleffects::slopes()`.

Value

A `marginaleffects` data frame.

`morie_effects_tidy` *Tidy a model with* **broom** *(fallback:* `summary()` *coefficients)*

Description

Thin extender over `broom::tidy()`. When **broom** is not installed, falls back to building a tidy-style data frame from `summary(model)$coefficients`, which is sufficient for the core `term` / `estimate` / `std.error` / `statistic` / `p.value` columns on the model classes (`lm` / `glm`) that rmorie ships.

Usage

```
morie_effects_tidy(model, ...)
```

Arguments

`model`	A fitted model object.
`...`	Further arguments forwarded to `broom::tidy()`.

Value

A data frame with one row per model term.

morie_eg_coint *Engle-Granger two-step cointegration test*

Description

Engle-Granger two-step cointegration test

Usage

```
morie_eg_coint(y1, y2, max_lag = NULL)
```

Arguments

y1	Numeric, first series.
y2	Numeric, second series.
max_lag	Max ADF augmentation lags. Default `floor(12*(n/100)^0.25)`.

Value

Named list with `adf_statistic`, `p_value`, `beta`, `n`, `method`.

Examples

```
morie_eg_coint(y1 = rnorm(100), y2 = rnorm(100))
```

morie_egarch_model *EGARCH(1,1) asymmetric volatility model*

Description

EGARCH(1,1) asymmetric volatility model

Usage

```
morie_egarch_model(x)
```

Arguments

x	Numeric return series.

Value

Named list with `omega`, `alpha`, `gamma`, `beta`, `loglik`, `conditional_variance`, `n`, `method`.

Examples

```
morie_egarch_model(x = rnorm(50))
```

`morie_ensure_extras` *Ensure optional packages are installed (interactive helper)*

Description

Function-body helper for morie endpoints that require optional `Suggests:` packages. If every package in `pkgs` is already installed, returns silently. Otherwise:

Usage

```
morie_ensure_extras(pkgs, ask = interactive(), repos = NULL)
```

Arguments

`pkgs`	Character vector of required package names.
`ask`	Logical. If `TRUE` (default), prompt in interactive sessions. Pass `FALSE` to skip the prompt and behave like non-interactive: error out with the install-hint message.
`repos`	Optional CRAN repo URL(s). Default uses `getOption("repos")`, falling back to RStudio CRAN.

Details

- In an **interactive** session, prompts the user once, and on consent installs the missing packages via `utils::install.packages()`.
- In a **non-interactive** session (R CMD check, CI, Rscript), throws an informative error with the morie command the user should run to fix things: `morie_install_extras(c("X", "Y"))`.

Why not auto-install silently? CRAN policy forbids packages from writing to the user's library or making network calls at function call time without explicit consent. The interactive prompt is the CRAN-blessed escape: the user IS the one consenting, and during R CMD check or any non-interactive run, the function refuses to touch the library.

Typical use inside a morie function body that needs DoubleML and ranger:

```
morie_estimate_irm <- function(...) {
  morie_ensure_extras(c("DoubleML", "ranger"))
  ...
}
```

Value

Invisibly `TRUE` if all `pkgs` are (now) installed; throws otherwise.

See Also

`morie_install_extras()` for the user-facing bulk installer.

Examples

```
## Not run:
  # Interactive (RStudio / R console): prompts to install if needed
  morie_ensure_extras(c("DoubleML", "ranger"))

  # CI / Rscript: errors with install-hint instead of installing
  morie_ensure_extras(c("DoubleML"), ask = FALSE)

## End(Not run)
```

---

morie_entheo_analyze_subject

Per-subject DMT vs PCB BOLD pipeline

Description

R parity of `morie.entheo_dmt.analyze_subject`. Runs the Layer-2 BOLD analyses (global-signal LZ complexity and dynamic functional connectivity) on one subject under each condition and returns a RichResult-style comparison summary.

Usage

```
morie_entheo_analyze_subject(
  subject_id,
  conditions = c("DMT", "PCB"),
  window = 30L,
  step = 5L
)
```

Arguments

`subject_id`	integer subject ID.
`conditions`	character vector. Conditions to evaluate.
`window`	integer dFC window (TRs).
`step`	integer dFC stride (TRs).

Value

RichResult-style named list with `payload$rows` as a per-condition list of result rows.

`morie_entheo_available_subjects`
DMT_Imaging: list motion-survived subject IDs

Description

R parity of `morie.entheo_dmt.available_subjects`. Scans the `fMRI/` directory of the on-disk DMT_Imaging mirror for `LongS{NN}{DMT,PCB}.mat` filenames and returns the integer IDs.

Usage

```
morie_entheo_available_subjects()
```

Value

integer vector of subject IDs sorted ascending. Empty if the dataset root is missing or the `fMRI/` folder is absent.

References

Timmermann, C. et al. (2023). Human brain effects of DMT assessed via EEG-fMRI. PNAS 120(13): e2218949120.

`morie_entheo_clone_dmt_imaging`
Clone the DMT_Imaging dataset (Timmerman et al.) into a local cache.

Description

`morie_entheo_clone_dmt_imaging()` shells out to `git clone` to fetch the open-source DMT_Imaging dataset published by Christopher Timmerman's group at https://github.com/timmer500/DMT_Imaging. After the clone completes, `load_dmt_imaging()` and `morie_entheo_load_*()` will pick the real fixture up automatically (they probe `$MORIE_DMT_IMAGING_ROOT` and the cache dir).

Usage

```
morie_entheo_clone_dmt_imaging(root = NULL, overwrite = FALSE, branch = NULL)
```

Arguments

`root`	Optional destination directory. Defaults to `$MORIE_DMT_IMAGING_ROOT`, else `file.path(morie_cache_dir(), "DMT_Imaging")`.
`overwrite`	Logical; if `TRUE` and `root` already exists, wipe it first (rare; defaults `FALSE`).
`branch`	Optional branch / tag / SHA to check out after clone. `NULL` uses the default upstream branch.

Details

This is opt-in – the package never auto-clones at load time. We clone from a specific commit by default to make tests reproducible; pass `branch = NULL` to track main.

Related upstream resources Vee surfaced 2026-05-25:

- https://github.com/timmer500/DMT_Imaging.git (the actual EEG+fMRI dataset)
- https://github.com/pnk314/psychedelics.git (analysis pipeline)
- https://github.com/lisagirard/Psychedelics.git (review repository)
- https://github.com/kianenigma/awesome-psychedelics.git (curated index)

Value

Invisibly returns the destination path.

`morie_entheo_dataset_overview`

DMT_Imaging: dataset overview

Description

R parity of `morie.entheo_dmt.dataset_overview`. Returns a RichResult-style summary of the on-disk DMT_Imaging mirror.

Usage

```
morie_entheo_dataset_overview()
```

Value

named list with `title`, `summary_lines`, `interpretation`, `payload`.

`morie_entheo_dynamic_functional_connectivity`
Sliding-window dynamic functional connectivity (dRSFC)

Description

R parity of `morie.entheo_dmt.dynamic_functional_connectivity`. For an AAL-parcellated BOLD matrix of shape (n_regions, n_TRs), computes the upper-triangular Pearson correlation matrix in each sliding window of `window` TRs advanced by `step` TRs. Mirrors the '`dRSFC.m`' Matlab script in '`DMT_Imaging/Scripts/`'.

Usage

```
morie_entheo_dynamic_functional_connectivity(bold, window = 30L, step = 5L)
```

Arguments

`bold`	numeric matrix (n_regions, n_TRs).
`window`	integer window length (TRs). Default 30.
`step`	integer window stride (TRs). Default 5.

Value

RichResult-style named list with per-window mean / std of the upper-triangular correlation vector.

References

Allen, E. A. et al. (2014). Tracking whole-brain connectivity dynamics in the resting state. Cereb. Cortex 24(3): 663-676.

`morie_entheo_load_eeg_region`
DMT_Imaging: load IRASA EEG regressors for one cortical region

---

Description

R parity of `morie.entheo_dmt.load_eeg_region`. Reads `RegressorsInterpscrubbedIRASA_<region>.mat` and returns the DMT, PCB, and difference regressor cubes.

Usage

```
morie_entheo_load_eeg_region(region)
```

Arguments

`region`	character. One of Central, Frontal, Occipital, Parietal, Temporal.

Value

named list with elements `regDMT`, `regPCB`, `regdiff`; each is a 3-D array of shape (14 subj, 840 TRs, 5 bands).

`morie_entheo_load_fmri_subject`

DMT_Imaging: load one subject's BOLD AAL parcellation

Description

R parity of `morie.entheo_dmt.load_fmri_subject`. Reads `LongS{NN}{DMT|PCB}.mat` and extracts the `BOLD_AAL` matrix (112 AAL regions x 840 TRs).

Usage

```
morie_entheo_load_fmri_subject(subject_id, condition = "DMT")
```

Arguments

`subject_id`	integer subject ID (e.g. 1, 2, 14).
`condition`	character: "DMT" (default) or "PCB".

Value

numeric matrix of shape (112, 840).

`morie_entheo_lz_complexity`

Lempel-Ziv (LZ76) complexity of a binarised signal

Description

R parity of `morie.entheo_dmt.lz_complexity`. The DMT-vs-PCB contrast on LZ complexity is one of Timmermann 2023's headline findings: LZ rises under DMT, indicating increased neural-signal diversity.

Usage

```
morie_entheo_lz_complexity(signal, threshold = NULL)
```

Arguments

`signal`	numeric vector.
`threshold`	numeric or NULL. Binarisation threshold. NULL = median (the standard choice).

Value

RichResult-style named list with raw and length-normalised LZ.

References

Lempel, A. & Ziv, J. (1976). On the complexity of finite sequences. IEEE Trans. Inf. Theory 22(1): 75-81. Schartner, M. et al. (2015). Complexity of multi-dimensional spontaneous EEG decreases during propofol-induced general anaesthesia. PLOS ONE 10(8): e0133532.

`morie_entheo_spectral_band_power`

EEG band-power decomposition via Welch PSD

Description

R parity of `morie.entheo_dmt.spectral_band_power`. Wraps `rgpsd` (morie's Welch PSD; same algorithm as SciPy's `welch`) and integrates the PSD over each canonical band by the trapezoidal rule.

Usage

```
morie_entheo_spectral_band_power(
  signal,
  fs = 200,
  bands = .MORIE_ENTHEO_DEFAULT_BANDS,
  nperseg = NULL
)
```

Arguments

`signal`	numeric vector. A 1-D EEG time series.
`fs`	numeric. Sampling frequency in Hz. Default 200 Hz (Timmermann 2023 acquisition).
`bands`	list of `list(name=, lo=, hi=)` entries. Default = canonical delta/theta/alpha/beta/gamma.
`nperseg`	integer or NULL. Welch segment length. Defaults to `min(length(signal), max(64, 4*fs))` – 4-second segments at fs.

Value

RichResult-style named list. `payload$rows` carries per-band absolute and relative power.

References

Welch, P. (1967). The use of FFT for the estimation of power spectra. IEEE Trans. Audio Electroacoust. 15(2): 70-73. Rangayyan, R. M. & Krishnan, S. (2024). Biomedical Signal Analysis, 3rd ed., Ch. 5.

`morie_estimate_aipw` *Augmented IPW (AIPW) doubly-robust ATE estimator*

Description

Combines IPW and outcome regression corrections. Consistent if **either** the propensity model **or** the outcome model is correctly specified.

Usage

```
morie_estimate_aipw(
  data,
  treatment,
  outcome,
  covariates,
  propensity_col = NULL,
  outcome_model = c("linear", "logistic")
)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`propensity_col`	Optional: name of a pre-computed propensity score column.
`outcome_model`	Family for the outcome model: `"linear"` or `"logistic"`.

Details

The propensity step delegates to **WeightIt** when installed (via `morie_estimate_propensity_scores`). The outcome regression and the doubly-robust influence-function score are evaluated inline to preserve the closed-form SE used downstream. Where richer outputs are desired, `AIPW::AIPW` (with SuperLearner nuisance learners) is the canonical CRAN counterpart.

Value

Named list: `ate`, `se`, `ci_lower`, `ci_upper`, `n`.

Examples

```
set.seed(1)
df <- data.frame(t = rbinom(200, 1, 0.4), y = rnorm(200), x = rnorm(200))
morie_estimate_aipw(df, "t", "y", "x")
```

---

morie_estimate_atc *Estimate the Average Treatment Effect on the Controls (ATC)*

Description

Control units receive weight 1; treated units receive $w_i = (1 - \hat{e}(X_i))/\hat{e}(X_i)$.

Usage

```
morie_estimate_atc(data, treatment, outcome, covariates, propensity_col = NULL)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`propensity_col`	Optional: name of a pre-computed propensity score column.

Details

Propensity-score estimation delegates to **WeightIt** when installed (via `morie_estimate_propensity_scores`); the weighted-difference and influence-function SE run inline.

Value

Named list: `atc`, `se`, `ci_lower`, `ci_upper`, `n_control`.

Examples

```
set.seed(1)
df <- data.frame(t = rbinom(200, 1, 0.4), y = rnorm(200), x = rnorm(200))
morie_estimate_atc(df, "t", "y", "x")
```

`morie_estimate_ate` *Estimate the Average Treatment Effect (ATE) via Hajek IPW*

Description

The Hajek estimator uses stabilised IPW weights:

$$\widehat{ATE} = \bar{y}_1^w - \bar{y}_0^w$$

where $\bar{y}_t^w = \sum_{T_i=t} w_i Y_i / \sum_{T_i=t} w_i$ and $w_i = T_i/\hat{e}(X_i) + (1-T_i)/(1-\hat{e}(X_i))$.

Usage

```
morie_estimate_ate(data, treatment, outcome, covariates, propensity_col = NULL)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`propensity_col`	Optional: name of a pre-computed propensity score column.

Details

When **WeightIt** is installed the propensity step delegates to `WeightIt::weightit()`; otherwise the inline logistic regression is used. The Hajek difference and influence-function SE below are evaluated inline either way so the result list shape and the closed-form variance preserved.

Value

Named list: `ate`, `se`, `ci_lower`, `ci_upper`, `n`, `ess`.

Examples

```
set.seed(1)
df <- data.frame(
  t = rbinom(200, 1, 0.4),
  y = rnorm(200),
  x = rnorm(200)
)
morie_estimate_ate(df, "t", "y", "x")
```

`morie_estimate_att` *Estimate the Average Treatment Effect on the Treated (ATT)*

Description

Treated units receive weight 1; controls receive $w_i = \hat{e}(X_i)/(1 - \hat{e}(X_i))$.

Usage

```
morie_estimate_att(data, treatment, outcome, covariates, propensity_col = NULL)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`propensity_col`	Optional: name of a pre-computed propensity score column.

Details

Propensity-score estimation delegates to **WeightIt** when installed (via `morie_estimate_propensity_scores`); the weighted-difference and influence-function SE run inline.

Value

Named list: `att`, `se`, `ci_lower`, `ci_upper`, `n_treated`.

Examples

```
set.seed(2)
df <- data.frame(t = rbinom(200, 1, 0.4), y = rnorm(200), x = rnorm(200))
morie_estimate_att(df, "t", "y", "x")
```

`morie_estimate_cate` *Estimate per-unit Conditional Average Treatment Effects (CATE)*

Description

The **T-learner** fits separate outcome models on treated and control units, then predicts the counterfactual for each unit: $\widehat{CATE}_i = \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i)$.

Usage

```
morie_estimate_cate(
  data,
  treatment,
  outcome,
  covariates,
  propensity_col = NULL,
  outcome_model = c("linear", "logistic"),
  meta_learner = c("t_learner", "s_learner")
)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`propensity_col`	Optional: name of a pre-computed propensity score column.
`outcome_model`	Family for the outcome model: `"linear"` or `"logistic"`.
`meta_learner`	`"t_learner"` (default) or `"s_learner"`.

Details

The **S-learner** fits one model with treatment as a feature.

For random-forest CATE estimation prefer `grf::causal_forest` (richer heterogeneity, honest sample splitting).

Value

Numeric vector of per-unit CATE estimates.

Examples

```
morie_estimate_cate(
  data = data.frame(
    t = stats::rbinom(100, 1, 0.4),
    y = stats::rbinom(100, 1, 0.3), x1 = stats::rnorm(100),
    x2 = stats::rnorm(100)
  ), treatment = "t", outcome = "y",
  covariates = c("x1", "x2")
)
```

`morie_estimate_double_ml` *Estimate ATE via Double Machine Learning (Partially Linear Regression)*

Description

Implements Chernozhukov et al. (2018) double/debiased machine learning for the partially linear regression model. When the **DoubleML** R package is installed, delegates to `DoubleML::DoubleMLPLR` with random-forest nuisance learners. Otherwise falls back to a hand-rolled cross-fit ridge implementation: residualise Y and D on X via K-fold ridge, then regress the outcome residual on the treatment residual.

Usage

```
morie_estimate_double_ml(
  data,
  outcome,
  treatment,
  covariates,
  n_folds = 5L,
  n_rep = 1L,
  random_state = 42L
)
```

Arguments

`data`	A data frame with treatment, outcome, and covariate columns.
`outcome`	Name of the continuous outcome column.
`treatment`	Name of the (binary) treatment column.
`covariates`	Character vector of covariate column names.
`n_folds`	Number of cross-fitting folds (default 5).
`n_rep`	Number of repeated cross-fitting repetitions (DoubleML only; ignored by the ridge fallback). Default 1.
`random_state`	Integer seed for cross-fit folds and learners (default 42).

Value

Named list with elements `ate`, `se`, `ci_lower`, `ci_upper`, `n`, `method`.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1), C1–C68.

Examples

```
set.seed(1)
n <- 200
X <- matrix(rnorm(n * 3), n, 3)
d <- rbinom(n, 1, plogis(X[, 1]))
y <- 0.5 * d + X[, 1] + rnorm(n)
df <- data.frame(y = y, d = d, x1 = X[, 1], x2 = X[, 2], x3 = X[, 3])
morie_estimate_double_ml(df, "y", "d", c("x1", "x2", "x3"))
```

---

morie_estimate_g_computation

G-computation (outcome regression) ATE estimator

Description

Estimates the ATE by:

$$\widehat{ATE} = \frac{1}{n} \sum_i \left[\hat{\mu}_1(X_i) - \hat{\mu}_0(X_i) \right]$$

Usage

```
morie_estimate_g_computation(
  data,
  treatment,
  outcome,
  covariates,
  outcome_model = c("linear", "logistic")
)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`outcome_model`	Family for the outcome model: `"linear"` or `"logistic"`.

Details

Delegates the standardisation step to `stdReg::stdGlm()` when **stdReg** is installed; otherwise computes the contrast inline from a single `stats::glm()` fit with treatment-flipped counterfactual datasets.

Value

Named list: `ate`, `se`, `ci_lower`, `ci_upper`.

Examples

```
set.seed(1)
df <- data.frame(t = rbinom(200, 1, 0.4), y = rnorm(200), x = rnorm(200))
morie_estimate_g_computation(df, "t", "y", "x")
```

---

morie_estimate_gate *Estimate Group Average Treatment Effects (GATE)*

Description

Applies AIPW within each level of `group_col` to estimate stratum-specific treatment effects.

Usage

```
morie_estimate_gate(
  data,
  treatment,
  outcome,
  covariates,
  group_col,
  propensity_col = NULL,
  outcome_model = c("linear", "logistic")
)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the outcome column.
`covariates`	Character vector of covariate names.
`group_col`	Name of the grouping variable (e.g. `"gender"`).
`propensity_col`	Optional: name of a pre-computed propensity score column.
`outcome_model`	Family for the outcome model: `"linear"` or `"logistic"`.

Value

Data frame with columns: `group`, `ate`, `se`, `ci_lower`, `ci_upper`, `n`.

Examples

```
set.seed(3)
df <- data.frame(
  t = rbinom(300, 1, 0.4),
  y = rnorm(300),
  x = rnorm(300),
```

```
  g = sample(c("A", "B"), 300, replace = TRUE)
)
morie_estimate_gate(df, "t", "y", "x", "g")
```

`morie_estimate_irm` *Estimate ATE via the Interactive Regression Model (IRM)*

Description

Implements the IRM variant of Chernozhukov et al. (2018) double machine learning, which allows treatment-effect heterogeneity by fitting separate outcome regressions for $T = 0$ and $T = 1$ alongside a propensity model. Uses `DoubleML::DoubleMLIRM` when available; otherwise falls back to a hand-rolled cross-fit estimator using logistic regression for the propensity score and ridge regression for the conditional outcome regressions.

Thin R wrapper that dispatches to the CRAN `DoubleML` package's `DoubleML::DoubleMLIRM` R6Class, mirroring the Python sibling `morie.estimate_irm()` (which dispatches to the Python `DoubleML` package).

Usage

```
morie_estimate_irm(
  data,
  treatment,
  outcome,
  covariates,
  n_folds = 5,
  random_state = 42
)

morie_estimate_irm(
  data,
  treatment,
  outcome,
  covariates,
  n_folds = 5,
  random_state = 42
)
```

Arguments

`data`	A `data.frame` containing outcome, treatment, and covariates.
`treatment`	Column name of the binary treatment.
`outcome`	Column name of the outcome.
`covariates`	Character vector of covariate column names.
`n_folds`	Number of cross-fitting folds (default 5).
`random_state`	Random seed (default 42).

Details

Following the DoubleML R package's own conventions, this uses the `mlr3` ecosystem for the nuisance learners (`ml_g` for $E[Y|T,X]$ and `ml_m` for $P(T = 1|X)$). Defaults are `lrn("regr.lm")` and `lrn("classif.log_reg")`, which require nothing beyond `stats`. For higher-capacity defaults, install `ranger` and pass `lrn("regr.ranger")` / `lrn("classif.ranger")` via the underlying `DoubleML::DoubleMLIRM$new()` directly.

Following Chernozhukov et al. (2018), the IRM extends the partially linear model by allowing fully heterogeneous treatment effects:

$$Y = g_0(T, X) + U, \quad E[U|T, X] = 0$$

$$T = m_0(X) + V, \quad E[V|X] = 0$$

Value

Named list with `ate`, `se`, `ci_lower`, `ci_upper`, `n`, `method`.

A list with components: `ate`, `se`, `ci_lower`, `ci_upper`, `n`, `method` (`"IRM (DoubleML)"`).

CRAN Suggests

Requires the suggested packages `DoubleML`, `mlr3`, and `mlr3learners`. Install with `install.packages(c("DoubleM "mlr3", "mlr3learners"))`. If any are unavailable, the function raises an informative error.

References

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1), C1–C68.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1), C1–C68. doi:10.1111/ectj.12097

Bach, P., Chernozhukov, V., Kurz, M. S., & Spindler, M. (2024). DoubleML – An object-oriented implementation of double machine learning in R. *Journal of Statistical Software*, 108(3). doi:10.18637/jss.v108.i03

Examples

```
set.seed(1)
n <- 200
X <- matrix(rnorm(n * 3), n, 3)
d <- rbinom(n, 1, plogis(X[, 1]))
y <- 0.5 * d + X[, 1] + rnorm(n)
df <- data.frame(y = y, d = d, x1 = X[, 1], x2 = X[, 2], x3 = X[, 3])
morie_estimate_irm(df, treatment = "d", outcome = "y",
                   covariates = c("x1", "x2", "x3"))

if (requireNamespace("DoubleML", quietly = TRUE) &&
  requireNamespace("mlr3", quietly = TRUE) &&
```

```
  requireNamespace("mlr3learners", quietly = TRUE)) {
  set.seed(1)
  n <- 200
  X <- matrix(rnorm(n * 5), n, 5)
  ps <- plogis(X[, 1] - X[, 2])
  T <- rbinom(n, 1, ps)
  Y <- 0.5 * T + X[, 1] + rnorm(n)
  df <- data.frame(Y = Y, T = T, X)
  morie_estimate_irm(df,
    treatment = "T", outcome = "Y",
    covariates = paste0("X", 1:5)
  )
}
```

`morie_estimate_late` *Estimate the Local Average Treatment Effect (LATE) via 2SLS / Wald*

Description

Uses a binary instrument Z to identify the LATE (Imbens & Angrist, 1994):

$$LATE = \frac{Cov(Y, Z)}{Cov(T, Z)}$$

Usage

```
morie_estimate_late(data, treatment, outcome, instrument, covariates = NULL)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary endogenous treatment column.
`outcome`	Name of the outcome column.
`instrument`	Name of the binary instrument column.
`covariates`	Optional character vector of exogenous covariates.

Details

With covariates, the routine delegates to `ivreg::ivreg()` (**ivreg**) or `AER::ivreg()` when either package is installed; otherwise it falls back to a manual two-stage OLS. Without covariates the closed-form Wald estimator and its delta-method SE are used.

Value

Named list: `late`, `se`, `ci_lower`, `ci_upper`, `first_stage_f`, `n`.

References

Imbens GW, Angrist JD (1994). Identification and estimation of local average treatment effects. *Econometrica*, 62(2), 467-475.

Examples

```
set.seed(1)
n <- 300L
z <- rbinom(n, 1, 0.5)
t <- rbinom(n, 1, plogis(-0.2 + 1.5 * z))
y <- 0.8 * t + rnorm(n)
morie_estimate_late(data.frame(t = t, y = y, z = z), "t", "y", "z")
```

---

morie_estimate_propensity_scores

Estimate propensity scores via logistic regression

Description

Thin wrapper over `WeightIt::weightit(method = "glm",estimand = "ATE")` when **WeightIt** is installed; falls back to `stats::glm(family = binomial())` otherwise.

Usage

```
morie_estimate_propensity_scores(
  data,
  treatment,
  covariates,
  trim = c(0.01, 0.99)
)
```

Arguments

`data`	A data frame.
`treatment`	Name of the binary treatment column.
`covariates`	Character vector of covariate names.
`trim`	Quantile pair used to winsorize extreme scores (default 0.01, 0.99).

Value

Numeric vector of propensity scores (same length as `nrow(data)`).

Examples

```
df <- data.frame(t = c(0, 1, 0, 1, 0, 1), x = rnorm(6))
ps <- morie_estimate_propensity_scores(df, "t", "x")
```

`morie_ewma_volatility` *EWMA volatility (RiskMetrics 1996)*

Description

EWMA volatility (RiskMetrics 1996)

Usage

```
morie_ewma_volatility(x, lambda = 0.94)
```

Arguments

`x`	Numeric return series.
`lambda`	Decay factor in (0,1). Default 0.94 (daily RiskMetrics).

Value

Named list with `conditional_variance`, `conditional_volatility`, `lambda`, `n`, `last_variance`, `last_volatility`, `method`.

Examples

```
morie_ewma_volatility(x = rnorm(50))
```

`morie_fairness_apply_profile` *Rename a city data.frame onto the canonical audit schema*

Description

Rename a city data.frame onto the canonical audit schema

Usage

```
morie_fairness_apply_profile(df, profile)
```

Arguments

`df`	A `data.frame` in the city's native column names.
`profile`	A `morie_city_profile` or the name (character) of a registered profile.

Value

A new `data.frame` with the profile's columns renamed to the canonical names, retaining only those canonical columns.

Examples

```
df <- data.frame(beat = c("A", "B"), score = c(0.1, 0.9))
p <- morie_fairness_city_profile(
  "demo", area_col = "beat", risk_col = "score"
)
morie_fairness_apply_profile(df, p)
```

`morie_fairness_average_odds_difference`
Average Odds Difference (mean of TPR and FPR gaps)

Description

For each non-reference group, the average odds difference is `0.5 * ((FPR_group - FPR_ref) + (TPR_group - TPR_ref))`. Zero means parity of errors; values away from zero mean the combined error profile favours one group over another. Used in IBM AIF360 and in the COMPAS *XAI Stories* audit.

For each non-reference group, `0.5 * ((FPR_group - FPR_ref) + (TPR_group - TPR_ref))`. Zero means parity of errors. This is the single-number summary used in IBM AIF360 and the COMPAS *XAI Stories* audit.

Usage

```
morie_fairness_average_odds_difference(
  y_true,
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)

morie_fairness_average_odds_difference(
  y_true,
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

Arguments

`y_true`	Vector of realised ground-truth outcomes.
`y_pred`	Vector of system decisions.
`group`	Vector of protected-attribute values.
`privileged`	The reference group (inferred if `NULL`).
`favorable`	The value treated as the positive class (default `1`).

Value

A `morie_fairness_result`; headline value is the largest absolute AOD across groups.

A named list: `value` (largest absolute AOD), `average_odds_difference`, `rates`, `privileged`, `warnings`, `interpretation`.

Examples

```
truth <- c(1, 0, 1, 0, 1, 0, 1, 0)
pred <- c(1, 0, 1, 0, 1, 1, 0, 1)
race <- c(rep("A", 4), rep("B", 4))
res <- morie_fairness_average_odds_difference(truth, pred, race,
  privileged = "A"
)
res$value # 0.25
```

---

morie_fairness_bias_amplification

Bias Amplification Score (parity gap x Gini of group rates)

Description

`BAS = delta_parity * G`, where `delta_parity` is the demographic-parity gap of the worst-affected group and `G` is the Gini coefficient of the per-group favourable-outcome rates. Large only when a directional disparity coincides with high cross-group inequality.

`BAS = Delta_parity * G`, where `Delta_parity` is the demographic parity gap of the worst-affected group and `G` is the Gini coefficient of the per-group favourable-outcome rates. Large only when a directional disparity coincides with high overall inequality.

Usage

```
morie_fairness_bias_amplification(
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)

morie_fairness_bias_amplification(
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

Arguments

`y_pred`	Vector of decisions/assignments, one per individual.
`group`	Vector of protected-attribute values (e.g. race).
`privileged`	The reference group. If `NULL` (default) the highest-rate group is used and a warning is attached.
`favorable`	The value of `y_pred` counted as the favourable outcome (default `1`).

Details

Reimplemented from Barman & Barman, arXiv:2603.18987.

Reimplemented from Barman & Barman, "Unmasking Algorithmic Bias in Predictive Policing" (arXiv:2603.18987).

Value

A `morie_fairness_result`; headline value is BAS.

A named list: `value` (BAS), `bias_amplification_score`, `demographic_parity_gap`, `gini`, `rates`, `privileged`, `warnings`, `interpretation`.

Examples

```
pred <- c(1, 1, 1, 1, 0, 0, 0, 0)
race <- c(rep("A", 4), rep("B", 4))
res <- morie_fairness_bias_amplification(pred, race, privileged = "A")
res$value # -0.5  (parity gap -1.0 times Gini 0.5)
```

MORIE_FAIRNESS_CANONICAL_FIELDS

The five canonical per-area fields the audit consumes.

Description

The five canonical per-area fields the audit consumes.

Usage

```
MORIE_FAIRNESS_CANONICAL_FIELDS
```

Format

An object of class `character` of length 5.

`morie_fairness_city_profile` *Construct a city profile (column map onto the canonical audit schema)*

Description

Each `*_col` argument names the column, in that city's own export, that carries the corresponding canonical field. `risk_col` and `outcome_col` may be `NULL` when a city only supplies one side (e.g. risk scores but no realised- outcome feed); the missing side must then be supplied separately to the audit.

Usage

```
morie_fairness_city_profile(
  name,
  area_col,
  risk_col = NULL,
  outcome_col = NULL,
  population_col = NULL,
  group_col = NULL,
  notes = ""
)
```

Arguments

`name`	Character identifier used with `morie_fairness_get_city`.
`area_col`	Column holding the area / district / precinct identifier. Required.
`risk_col`, `outcome_col`, `population_col`, `group_col`	Optional columns for predicted risk, realised-outcome count, area population, and protected attribute.
`notes`	Free-text provenance or caveats, surfaced to the user.

Value

A list of class `morie_city_profile`.

Examples

```
p <- morie_fairness_city_profile(
  "chicago", area_col = "community_area",
  risk_col = "rti", group_col = "majority_race"
)
p$name
```

morie_fairness_column_map

Column map for a city profile

Description

Returns a named character vector mapping source column names (those defined on the profile) to canonical field names.

Usage

```
morie_fairness_column_map(profile)
```

Arguments

profile	A morie_city_profile.

Value

A named character vector c(source = canonical).

morie_fairness_demographic_parity

Demographic Parity Gap (difference in favourable-outcome rates)

Description

The gap is rate(group) - rate(privileged). Demographic parity holds when every group receives favourable outcomes at the same rate, i.e. all gaps are zero. Unlike the disparate-impact ratio this additive form is well defined even when the privileged rate is zero.

The additive difference in favourable-outcome rates, rate(group) - rate(privileged). Demographic parity holds when every group receives favourable outcomes at the same rate.

Usage

```
morie_fairness_demographic_parity(
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)

morie_fairness_demographic_parity(
  y_pred,
  group,
```

```
  privileged = NULL,
  favorable = 1
)
```

Arguments

`y_pred`	Vector of decisions/assignments, one per individual.
`group`	Vector of protected-attribute values (e.g. race).
`privileged`	The reference group. If `NULL` (default) the highest-rate group is used and a warning is attached.
`favorable`	The value of `y_pred` counted as the favourable outcome (default `1`).

Value

A `morie_fairness_result`; headline value is the largest absolute gap across groups.

A named list: `value` (largest absolute gap), `gaps`, `rates`, `privileged`, `warnings`, `interpretation`.

Examples

```
pred <- c(1, 1, 1, 1, 0, 0, 0, 1, 0, 0)
race <- c("A","A","A","A","A","B","B","B","B","B")
morie_fairness_demographic_parity(pred, race, privileged = "A")$value
pred <- c(1, 1, 1, 1, 0, 0, 0, 1, 0, 0)
race <- c(rep("A", 5), rep("B", 5))
res <- morie_fairness_demographic_parity(pred, race, privileged = "A")
res$value # -0.6  (group B rate 0.2 minus group A rate 0.8)
```

---

morie_fairness_disparate_impact

Disparate Impact Ratio (EEOC four-fifths / 80% rule)

Description

For each group, the disparate-impact ratio is its favourable- outcome rate divided by the privileged group's rate. A value below 0.8 is the standard legal indicator of adverse impact.

For each group, the disparate-impact ratio is its favourable-outcome rate divided by the privileged group's rate. A value below 0.8 is the standard legal indicator of adverse impact.

Usage

```
morie_fairness_disparate_impact(
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

```
morie_fairness_disparate_impact(
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

Arguments

`y_pred`	Vector of decisions/assignments, one per individual.
`group`	Vector of protected-attribute values (e.g. race).
`privileged`	The reference group. If `NULL` (default) the highest-rate group is used and a warning is attached.
`favorable`	The value of `y_pred` counted as the favourable outcome (default `1`).

Value

A `morie_fairness_result`; headline value is the worst (smallest) ratio across groups.

A named list: `value` (worst ratio), `ratios`, `rates`, `privileged`, `adverse_impact`, `threshold`, `warnings`, `interpretation`.

Examples

```
pred <- c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0)
race <- c("A","A","A","A","A","B","B","B","B","B")
morie_fairness_disparate_impact(pred, race, privileged = "A")$value
pred <- c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0)
race <- c(rep("A", 5), rep("B", 5))
res <- morie_fairness_disparate_impact(pred, race, privileged = "A")
res$value # 0.6  (group B rate 0.6 / group A rate 1.0)
res$adverse_impact # TRUE
```

`morie_fairness_equalized_odds` *Equalized Odds (TPR / FPR gaps across groups)*

Description

Equalized odds holds when both the true-positive rate and the false-positive rate are equal across groups. Needs ground truth, so it audits a system's errors rather than its decision rates - a system can satisfy demographic parity yet make many more false positives against one group.

Equalized odds holds when the true-positive rate (TPR) and false-positive rate (FPR) are equal across groups. Needs ground-truth labels, so it audits a system's *errors*, not just its decision rates.

Usage

```
morie_fairness_equalized_odds(
  y_true,
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)

morie_fairness_equalized_odds(
  y_true,
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

Arguments

`y_true`	Vector of realised ground-truth outcomes.
`y_pred`	Vector of system decisions.
`group`	Vector of protected-attribute values.
`privileged`	The reference group (inferred if `NULL`).
`favorable`	The value treated as the positive class (default `1`).

Value

A `morie_fairness_result`; headline value is the largest absolute TPR-or-FPR gap.

A named list: `value` (largest absolute TPR/FPR gap), `tpr_gaps`, `fpr_gaps`, `rates`, `privileged`, `violation`, `warnings`, `interpretation`.

Examples

```
truth <- c(1,0,1,0,1,0,1,0)
pred  <- c(1,0,1,0,1,1,0,1)
race  <- c("A","A","A","A","B","B","B","B")
morie_fairness_equalized_odds(truth, pred, race, privileged="A")$value
truth <- c(1, 0, 1, 0, 1, 0, 1, 0)
pred <- c(1, 0, 1, 0, 1, 1, 0, 1)
race <- c(rep("A", 4), rep("B", 4))
res <- morie_fairness_equalized_odds(truth, pred, race, privileged = "A")
res$violation # TRUE
```

morie_fairness_get_city

Look up a registered city profile by case-insensitive name

Description

Look up a registered city profile by case-insensitive name

Usage

```
morie_fairness_get_city(name)
```

Arguments

`name`	Character. The profile name (case-insensitive).

Value

A `morie_city_profile`.

morie_fairness_gini

Gini coefficient (concentration of a non-negative distribution)

Description

Ranges from 0 (perfect equality) to nearly 1 (one unit holds everything). Applied to risk scores or stop counts, it measures how unequally a predictive system concentrates its attention. When `group` is supplied, per-group Gini values are also reported.

Ranges from 0 (perfect equality) to nearly 1 (one unit holds everything). Applied to risk scores or patrol counts it measures how unequally a system concentrates its attention. With `group` supplied, a per-group Gini is also reported.

Usage

```
morie_fairness_gini(values, group = NULL)

morie_fairness_gini(values, group = NULL)
```

Arguments

`values`	Vector of non-negative quantities.
`group`	Optional vector of protected-attribute values; enables the per-group breakdown.

Value

A `morie_fairness_result`; headline value is the overall Gini.

A named list: `value` (overall Gini), `gini`, `per_group`, `warnings`, `interpretation`.

Examples

```
morie_fairness_gini(c(5, 5, 5, 5))$value
morie_fairness_gini(c(0, 0, 0, 100))$value
morie_fairness_gini(c(5, 5, 5, 5))$value # 0
morie_fairness_gini(c(0, 0, 0, 100))$value # 0.75
```

`morie_fairness_list_cities` *List registered city profile names*

Description

List registered city profile names

Usage

```
morie_fairness_list_cities()
```

Value

Sorted character vector of registered profile names.

`morie_fairness_noisy_or_detection` *Noisy-OR patrol-detection probabilities*

Description

For each crime location, computes `1 - (1 - p_detect)^k` where `k` is the number of officers within `radius`.

Usage

```
morie_fairness_noisy_or_detection(
  crime_xy,
  officer_xy,
  radius,
  p_detect = 0.85,
  seed = NULL
)
```

Arguments

`crime_xy`	Numeric (n, 2) matrix of crime coordinates.
`officer_xy`	Numeric (m, 2) matrix of officer coordinates.
`radius`	Detection radius (positive).
`p_detect`	Per-officer detection probability in (0, 1].
`seed`	Optional integer; when supplied, a Bernoulli outcome is sampled per crime and returned in `$detected`.

Value

`morie_fairness_result` with `$probabilities`, `$officers_in_range`, optional `$detected`.

`morie_fairness_predpol` *Predictive-policing disparity audit (port of morie.fairness.predpol)*

Description

Three callables that mirror the SciencesPo Predictive-Policing-Chicago analysis, city-agnostically:

Details

- `morie_fairness_predpol_aggregate_areas`: per-record to per-area roll-up.
- `morie_fairness_predpol_calibration_audit`: Spearman correlation between predicted risk rank and realised outcome rank, plus per-group rank-gap analysis.
- `morie_fairness_predpol_score_disparity`: descriptive score-by-group summary with one-way ANOVA.

`morie_fairness_predpol_aggregate_areas` *Aggregate per-record predictive-policing data to per-area*

Description

Aggregate per-record predictive-policing data to per-area

Usage

```
morie_fairness_predpol_aggregate_areas(
  area,
  risk,
  outcome,
  group = NULL,
  population = NULL
)
```

Arguments

`area`	Area identifier per record.
`risk`	Predicted-risk score per record.
`outcome`	Realised-outcome indicator/count per record.
`group`	Optional protected attribute per record; the per-area majority becomes the area's label.
`population`	Optional named numeric vector mapping area to population, or a per-record vector (taken as constant within an area). When supplied, outcome rate is per 10,000 inhabitants.

Value

A list with `areas`, `mean_risk`, `outcome_rate`, `group`, `n_records`.

`morie_fairness_predpol_calibration_audit`

Predicted-vs-realised rank audit by demographic group

Description

Predicted-vs-realised rank audit by demographic group

Usage

```
morie_fairness_predpol_calibration_audit(areas, mean_risk, outcome_rate, group)
```

Arguments

`areas`	Area identifiers (per area, not per record).
`mean_risk`	Mean predicted-risk score per area.
`outcome_rate`	Realised-outcome rate per area.
`group`	Majority/dominant group per area.

Value

`morie_fairness_result`; `$value` is the largest-magnitude per-group mean rank gap (positive = over-policed).

morie_fairness_predpol_score_disparity

Descriptive score-by-group disparity

Description

Descriptive score-by-group disparity

Usage

```
morie_fairness_predpol_score_disparity(score, group, reference = NULL)
```

Arguments

`score`	Continuous risk score per individual.
`group`	Protected attribute per individual.
`reference`	Optional reference group label (default: lowest-mean).

Value

`morie_fairness_result`; `$value` is the spread (max - min) of per-group mean scores.

morie_fairness_predpol_temporal_audit

Audit how disparity metrics move over time and across cities

Description

For every (`city`, `period`) cell the audit computes the four disparity metrics, then aggregates per city - reporting the mean of each metric, the count of periods with DIR > 1 (over-prediction periods), and the DIR temporal range (max - min) which quantifies how unstable the metric is across the audited window.

Usage

```
morie_fairness_predpol_temporal_audit(
  period,
  city,
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

Arguments

`period`	Time-period label per record (e.g. `"2019-03"`). Sorted lexically for display, so ISO-style labels order correctly.
`city`	City label per record.
`y_pred`	Decision / assignment per record.
`group`	Protected attribute per record.
`privileged`	Reference group. If `NULL`, inferred globally from pooled data.
`favorable`	Value of `y_pred` counted as favourable (default `1`).

Details

The reference (privileged) group is inferred **globally** from the pooled data when not supplied, so every cell uses the same reference.

Value

A `morie_fairness_result`; headline value is the largest per-city DIR temporal range - the worst temporal instability found in the audited window.

Examples

```
period <- c(rep("p1", 10), rep("p2", 10))
city   <- rep("A", 20)
pred   <- rep(c(1,1,1,1,1,1,1,1,0,0), 2)
grp    <- rep(c(rep("X",5), rep("Y",5)), 2)
res <- morie_fairness_predpol_temporal_audit(
  period, city, pred, grp, privileged = "X"
)
res$payload$per_city$A$dir_range
```

`morie_fairness_register_city`

Register a city profile in the process-local registry

Description

Register a city profile in the process-local registry

Usage

```
morie_fairness_register_city(profile, overwrite = FALSE)
```

Arguments

`profile`	A `morie_city_profile`.
`overwrite`	If `FALSE` (default), registering an existing name raises an error; pass `TRUE` to replace it.

Value

Invisibly returns the registered profile.

`morie_fairness_simulate_biased_crime_data`
Synthetic predictive-policing dataset with a known disparity

Description

Generates per-record data with `area`, `group`, `true_outcome` (group-independent Bernoulli at `base_rate`), `detected` (group-dependent), and `risk_score` (0–500, shifted up by `bias * 100` points for non-reference groups). The `bias` input is the ground truth the audits should recover.

Usage

```
morie_fairness_simulate_biased_crime_data(
  n = 2000L,
  groups = c("A", "B"),
  group_props = NULL,
  n_areas = 20L,
  base_rate = 0.3,
  bias = 0.5,
  seed = 0L
)
```

Arguments

`n`	Number of records.
`groups`	Character vector of group labels (the first entry is treated as the reference group).
`group_props`	Optional sampling proportions.
`n_areas`	Number of areas (>= number of groups).
`base_rate`	Reference-group favourable-outcome rate in 0–1.
`bias`	Injected disparity in -1–1.
`seed`	Reproducibility seed.

Value

A data.frame with columns area, group, true_outcome, detected, risk_score.

`morie_fairness_simulation`
Simulation primitives for the predictive-policing audit subsystem

---

Description

Pure base-R ports of the Noisy-OR detection model and the biased crime-data simulator from `morie.fairness.simulation`, both originally distilled from Barman & Barman (arXiv:2603.18987). No optional dependencies.

`morie_fairness_xai` *Model-agnostic explainability (XAI) for bias discovery*

Description

R ports of the explainer suite in `morie.fairness.xai`. Prefers **iml** for permutation importance / PDP / SHAP-ish attributions when available; otherwise computes the same quantities in base R from first principles. Every callable takes a `predict_fn` closure (matrix -> numeric vector) so it works on any classifier or risk model.

Details

- `morie_fairness_xai_permutation_importance`
- `morie_fairness_xai_partial_dependence`
- `morie_fairness_xai_ale`
- `morie_fairness_xai_ceteris_paribus`
- `morie_fairness_xai_shap_values`

`morie_fairness_xai_ale`
First-order Accumulated Local Effects (Apley & Zhu)

Description

First-order Accumulated Local Effects (Apley & Zhu)

Usage

```
morie_fairness_xai_ale(
  predict_fn,
  X,
  feature,
  feature_names = NULL,
  n_bins = 10L
)
```

Arguments

`predict_fn`	Function mapping an (n, d) matrix to n numeric predictions.
`X`	Numeric matrix or data.frame.
`feature`	Index or name of the feature to sweep.
`feature_names`	Optional character vector.
`n_bins`	Number of quantile bins.

Value

`morie_fairness_result`; `$value` is the ALE range.

`morie_fairness_xai_ceteris_paribus`

Ceteris-paribus profile for one instance

Description

Holds every feature of `x` fixed except `feature`, sweeps it across the range observed in `X_ref`, and reports the resulting prediction profile.

Usage

```
morie_fairness_xai_ceteris_paribus(
  predict_fn,
  x,
  feature,
  X_ref,
  feature_names = NULL,
  grid_size = 20L
)
```

Arguments

`predict_fn`	Function mapping (n, d) matrix to n predictions.
`x`	Numeric vector of length d (the instance).
`feature`	Index or name of the feature to vary.
`X_ref`	Reference matrix used for the feature range.
`feature_names`	Optional character vector.
`grid_size`	Number of grid points.

Value

`morie_fairness_result`; `$value` is the profile's swing (max - min).

`morie_fairness_xai_partial_dependence`
Partial dependence on one feature (Friedman)

Description

Partial dependence on one feature (Friedman)

Usage

```
morie_fairness_xai_partial_dependence(
  predict_fn,
  X,
  feature,
  feature_names = NULL,
  grid_size = 20L
)
```

Arguments

`predict_fn`	Function mapping an (n, d) matrix to n numeric predictions.
`X`	Numeric matrix or data.frame.
`feature`	Index or name of the feature to sweep.
`feature_names`	Optional character vector.
`grid_size`	Number of grid points.

Value

`morie_fairness_result`; `$value` is the PD range.

`morie_fairness_xai_permutation_importance`
Permutation feature importance (model-agnostic)

Description

Permutation feature importance (model-agnostic)

Usage

```
morie_fairness_xai_permutation_importance(
  predict_fn,
  X,
  feature_names = NULL,
  n_repeats = 10L,
  protected = NULL,
  seed = 0L
)
```

Arguments

`predict_fn`	Function mapping an (n, d) matrix to n numeric predictions.
`X`	Numeric matrix or data.frame.
`feature_names`	Optional character vector.
`n_repeats`	Shuffles averaged per feature.
`protected`	Character vector of protected-attribute names; any that rank in the top third trigger a bias warning.
`seed`	Reproducibility seed.

Value

`morie_fairness_result`; `$value` is the largest importance.

`morie_fairness_xai_shap_values` *Shapley feature attributions for one instance (sampling estimator)*

Description

Shapley feature attributions for one instance (sampling estimator)

Usage

```
morie_fairness_xai_shap_values(
  predict_fn,
  x,
  background,
  feature_names = NULL,
  n_samples = 200L,
  seed = 0L
)
```

Arguments

`predict_fn`	Function mapping (n, d) matrix to n predictions.
`x`	Numeric vector of length d (the instance).
`background`	Reference matrix (n_bg, d) for marginal averaging.
`feature_names`	Optional character vector.
`n_samples`	Number of random permutations averaged.
`seed`	Reproducibility seed.

Value

`morie_fairness_result`; `$value` is the largest-magnitude SHAP value.

`morie_fast_available` *Is the R-side JIT acceleration active?*

Description

Mirrors `morie.fast.is_jit_available()` on the Python side. Returns TRUE when the Rcpp .so was built and loaded; FALSE when falling back to base-R implementations.

Usage

```
morie_fast_available()
```

Value

A logical scalar: `TRUE` when the compiled Rcpp backend was built and loaded, `FALSE` when falling back to base-R kernels.

Examples

```
morie_fast_available()
```

morie_fdr_qvalues — *FDR / q-values / tail-area p-values via* **fdrtool**

Description

Thin extender over `fdrtool::fdrtool` for the Strimmer (2008) shrinkage estimator of local and tail-area false-discovery rates, q-values, and the underlying null-distribution scale parameter.

Usage

```
morie_fdr_qvalues(x, statistic = "normal", ...)
```

Arguments

`x`	Numeric vector of test statistics or p-values, as appropriate to `statistic`.
`statistic`	Character; the type of statistic in `x`. One of `"normal"` (default), `"correlation"`, `"pvalue"`, or `"studentt"`.
`...`	Further arguments forwarded to `fdrtool::fdrtool` (e.g. `plot`, `color.figure`, `verbose`, `cutoff.method`, `pct0`).

Value

A list with `$method = "fdrtool::fdrtool"` and `$raw` (the `fdrtool` return list with `pval`, `qval`, `lfdr`, and `param`).

Examples

```
## Not run:
  if (requireNamespace("fdrtool", quietly = TRUE)) {
    set.seed(1)
    x <- c(stats::rnorm(900), stats::rnorm(100, mean = 3))
    morie_fdr_qvalues(x, statistic = "normal")
  }

## End(Not run)
```

morie_fetch — *Fetch a dataset from any URL, with automatic format detection*

Description

A universal data-access entry point. Given a URL, MORIE detects the format from the HTTP `Content-Type` header (falling back to the URL extension), downloads the resource, and parses it into an R object. The behaviour is automatic by default but every step is controllable: pass an explicit `format`, extra query `params`, a `zip_member` to extract, or reader arguments via `...`.

Usage

```
morie_fetch(
  url,
  format = c("auto", "csv", "tsv", "json", "xml", "html", "xlsx", "zip", "arcgis"),
  params = NULL,
  zip_member = "",
  simplify = TRUE,
  ...
)
```

Arguments

`url`	The resource URL.
`format`	One of `"auto"` (default), `"csv"`, `"tsv"`, `"json"`, `"xml"`, `"html"`, `"xlsx"`, `"zip"`, `"arcgis"`.
`params`	Optional named list appended to `url` as a URL-encoded query string.
`zip_member`	For `zip` downloads, the archive member to extract (matched by basename, then by substring).
`simplify`	For `json`/`xml`/`html`, whether to simplify into a data.frame where possible (default `TRUE`).
`...`	Passed to the underlying reader (e.g. `read.csv` arguments, or `morie_fetch_arcgis` arguments).

Details

Supported formats: `csv`, `tsv`, `json`, `xml`, `html`, `xlsx`, `zip` (extract one member), and `arcgis` (delegates to `morie_fetch_arcgis`).

Value

A data.frame for tabular formats; a list or document object for non-tabular `json`/`xml`/`html`.

See Also

`morie_ckan_search`, `morie_fetch_arcgis`

Examples

```
## Not run:
# Examples use placeholder URLs (example.org). Replace with a
# real CSV / JSON endpoint when running.
df <- morie_fetch("https://example.org/data.csv")
js <- morie_fetch("https://api.example.org/records",
  format = "json", params = list(limit = 100)
)

## End(Not run)
```

`morie_fetch_arcgis` *Query an ArcGIS FeatureServer / MapServer layer*

Description

Pulls attribute records from an ArcGIS REST layer, paginating through the server transfer limit automatically (ArcGIS caps a single query at `maxRecordCount` features, typically 1000-2000).

Usage

```
morie_fetch_arcgis(
  layer_url,
  where = "1=1",
  out_fields = "*",
  params = NULL,
  page_size = 2000L,
  max_records = Inf
)
```

Arguments

`layer_url`	The layer URL, ending in `/FeatureServer/<n>` or `/MapServer/<n>`.
`where`	SQL-style WHERE filter (default `"1=1"`, all rows).
`out_fields`	Comma-separated field list (default `"*"`).
`params`	Optional named list of extra query parameters.
`page_size`	Records requested per page (default 2000).
`max_records`	Cap on the total number of records (default `Inf` – fetch the whole layer).

Value

A data.frame of feature attributes (geometry is dropped).

See Also

morie_fetch

Examples

```
## Not run:
layer <- paste0(
  "https://services.arcgis.com/ORG/arcgis/rest/",
  "services/Assault/FeatureServer/0"
)
df <- morie_fetch_arcgis(layer)

## End(Not run)
```

`morie_fetch_ckan` *Fetch data from the CKAN API and cache it*

Description

Fetch data from the CKAN API and cache it

Usage

```
morie_fetch_ckan(
  dataset_key = "cpads",
  limit = Inf,
  db_path = NULL,
  resource_id = NULL,
  con = NULL
)
```

Arguments

`dataset_key`	One of `"cpads"`, `"csads"`, `"csus"`.
`limit`	Maximum records to fetch. The CKAN datastore caps a single request at 32000 rows, so larger resources are paged through with `offset`; the default reads the entire resource.
`db_path`	Optional override for the database path.
`resource_id`	Optional CKAN datastore resource id. When supplied (e.g. from `morie_dataset_catalog()$ck` it is used directly, so any catalogued dataset can be fetched without a built-in database; `dataset_key` then only labels the cache table.
`con`	Optional pre-opened DBI connection (overrides `db_path`).

Value

A data.frame.

Examples

```
## Not run:
# Requires network access. Fetches the first 5000 rows of the
# Canadian Postsecondary Alcohol and Drug Use Survey from the
# Government of Canada CKAN datastore:
cpads <- morie_fetch_ckan(dataset_key = "cpads", limit = 5000L)
nrow(cpads)

## End(Not run)
```

`morie_fetch_siu` *Fetch the Ontario SIU corpus into a 64-column SIU.csv*

Description

Fetches and parses the Ontario Special Investigations Unit (police-oversight) corpus – every director's report and the news releases they link – into a single CSV with the canonical 64-column schema, one row per case.

Usage

```
morie_fetch_siu(
  cache_dir = file.path(tempdir(), "morie", "siu"),
  overwrite = FALSE,
  max_drid = NULL,
  concurrency = 4L,
  rate_rps = 4,
  use_manifest = TRUE,
  lang = c("all", "en", "fr"),
  cache_html = FALSE,
  progress = TRUE
)
```

Arguments

`cache_dir`	Output directory. Defaults to a session-scoped subdirectory of `tempdir()` that R cleans up automatically. For persistent cross-session caching pass `cache_dir = morie_cache_dir("siu")` instead; see `morie_cache_dir` and `morie_cache_clear`.
`overwrite`	Logical; if `FALSE` and `SIU.csv` already exists in `cache_dir`, its path is returned without reparsing.
`max_drid`	Highest director's-report id to fetch. `NULL` (default) uses the shipped manifest's max + a small margin, falling back to discovery from the SIU site.
`concurrency`	Maximum simultaneous HTTP transfers. Default `4` is a polite rate paired with `rate_rps = 4`; raising either above ~8/8 risks triggering WAF interstitials that return short non-report HTML.
`rate_rps`	Maximum request starts per second across the pool (token-bucket throttle). Default `4` is the rate the package was empirically validated against; lower it on poor connections or contested endpoints.
`use_manifest`	If `TRUE` (default), restrict the sweep to the known-valid drids in the shipped manifest (`inst/extdata/siu_drid_manifest.csv.gz`), still topping up with any drid above the manifest's max up to `max_drid`. Cuts the fetch by ~30-50 percent on a typical run by skipping holes.

`lang` Language filter on the manifest. `"all"` (default, back-compat) fetches every known-valid drid – English and French copies of each case – and then collapses to one row per case_number with English winning the dedupe. `"en"` fetches only the English drids (about half the size of the corpus and half the network round trips); `"fr"` fetches only French. Use `"en"` for the fastest cold-start when you only need the canonical English text.

`cache_html` If `TRUE`, gzip and save the raw HTML of every fetched director's-report and news-release page under `<cache_dir>/html/drid_NNNN.html.gz` and `<cache_dir>/html/nrid_NNNN.html.gz`. This is the persistent ground truth for every row in the emitted CSV: any later discrepancy between the parser and a human coder can be adjudicated against the saved HTML without re-hitting SIU. Adds ~80-100 MB to `cache_dir` for a full run; default `FALSE` (the harvester remains lean unless you ask).

`progress` Logical; print progress messages.

Details

The parser is implemented entirely in C/C++ (`src/siu_parser.cpp`): libcurl drives the HTTP transport and a concurrent `curl_multi` pool fetches the ~9,000+ pages, while the 64-field extraction is C++ `std::regex` parsing. There is no Python dependency.

This is the *Ontario* Special Investigations Unit – distinct from the federal Structured Intervention Units and from OTIS. The parsed corpus is not shipped with the package; each user runs the parser themselves, which is fair use of public oversight reports.

Value

Path to the written `SIU.csv`.

Examples

```
## Not run:
# Network: parses the full Ontario SIU corpus (~15-25 min at the
# default polite rate of 4 RPS).
csv <- morie_fetch_siu(cache_dir = tempdir())
siu <- utils::read.csv(csv)
nrow(siu)

## End(Not run)
```

`morie_fetch_tps` *Fetch a TPS category from the Toronto Police ArcGIS REST endpoint*

Description

Pages through the ArcGIS `/query` endpoint and writes a tidy CSV to the morie cache directory. Calls back to a cached file on subsequent calls unless `overwrite = TRUE`.

Usage

```
morie_fetch_tps(
  category,
  cache_dir = file.path(tempdir(), "morie", "tps"),
  where = "1=1",
  overwrite = FALSE,
  max_per_page = 2000L
)
```

Arguments

`category`	One of `names(morie_tps_layer_urls())`.
`cache_dir`	Directory for the CSV. Defaults to a session-scoped subdirectory of `tempdir()` that R cleans up automatically. For persistent caching pass `cache_dir = morie_cache_dir("tps")`; see morie_cache_dir and morie_cache_clear.
`where`	ArcGIS SQL where clause (default `"1=1"`).
`overwrite`	Logical; if `FALSE` and the CSV exists, return its path without re-downloading.
`max_per_page`	ArcGIS page size (default 2000; server caps).

Value

Path to the CSV.

Examples

```
## Not run:
# Network: fetches major-crime indicators from the Toronto Police
# ArcGIS open-data layer.
csv <- morie_fetch_tps(
  category = "Assault",
  cache_dir = tempdir(),
  where = "OCC_YEAR = 2024"
)
tps <- utils::read.csv(csv)
nrow(tps)

## End(Not run)
```

---

morie_fisher_exact_test

Fisher's exact test for 2x2 tables

Description

Fisher's exact test for 2x2 tables

Usage

```
morie_fisher_exact_test(
  table_2x2,
  alternative = c("two.sided", "greater", "less")
)
```

Arguments

`table_2x2`	A 2x2 matrix or data frame of counts.
`alternative`	`"two.sided"`, `"greater"`, or `"less"`.

Value

Named list: `odds_ratio`, `ci`, `p_value`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_garch_fit` *Fit a GARCH(1,1) model to a return series*

Description

$$\sigma_t^2 = \omega + \alpha \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2.$$

Usage

```
morie_garch_fit(x)
```

Arguments

`x`	Numeric return series.

Value

Named list with `omega`, `alpha`, `beta`, `persistence`, `loglik`, `conditional_variance`, `n`, `method`.

Examples

```
morie_garch_fit(x = rnorm(50))
```

morie_gblup_full *GBLUP – full mixed-model implementation*

Description

Solves Henderson's MME with VanRaden G.

Usage

```
morie_gblup_full(x, y, markers, lambda_gblup = NULL)
```

Arguments

x	Fixed-effect design (vector or matrix).
y	Numeric response.
markers	Genotype matrix (n x m).
lambda_gblup	Optional ratio sigma_e^2 / sigma_g^2 (default h^2=0.5).

Value

Named list (estimate, g_hat, beta, se, lambda_gblup, n, method).

References

Montesinos Lopez Ch 3.

Examples

```
morie_gblup_full(x = rnorm(50), y = rnorm(50), markers = matrix(sample(0:2, 200, TRUE), 50, 4))
```

morie_generate_ar_coefficients *Generate a stationarity-preserving AR coefficient matrix*

Description

Generate a stationarity-preserving AR coefficient matrix

Usage

```
morie_generate_ar_coefficients(
  p,
  rng,
  spectral_radius = 0.8,
  diagonal_bias = 0.4
)
```

Arguments

p	Dimension (number of variables).
rng	An environment from morie_sync_rng().
spectral_radius	Target spectral radius < 1.
diagonal_bias	Mixture weight between diagonal autoregression (1) and full off-diagonal coupling (0).

Value

A p x p numeric matrix A.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

---

morie_generate_synthetic_data

Generate synthetic epidemiology-style tabular data

Description

Generates non-identifying synthetic data suitable for development, testing, and demos. The generator uses a canonical variable set and allows output column renaming through name_map so it can be adapted to multiple studies. Synthetic data should not be used for final inferential reporting.

Usage

```
morie_generate_synthetic_data(
  n = 5000L,
  seed = 42L,
  special_code_rate = 0.02,
  profile = c("generic", "morie_legacy"),
  name_map = NULL
)
```

Arguments

n	Number of rows.
seed	Random seed for reproducibility.
special_code_rate	Proportion of values replaced with survey-style special missing codes (97/98/99/997/998/999) in discrete fields.

`profile`	Convenience profile for output naming; ignored when `name_map` is supplied.
`name_map`	Optional named character vector mapping canonical keys to output column names. Use `morie_default_synthetic_name_map()` as a template.

Value

A data.frame with synthetic records.

Examples

```
df <- morie_generate_synthetic_data(n = 200, seed = 1)
nrow(df)
```

---

morie_generate_var_coefficients

Generate a VAR(L) coefficient array as a 3-d list

Description

Generate a VAR(L) coefficient array as a 3-d list

Usage

```
morie_generate_var_coefficients(
  p,
  lags,
  rng,
  spectral_radius = 0.8,
  decay = 0.6
)
```

Arguments

`p`	Number of variables.
`lags`	Number of lag matrices.
`rng`	`morie_sync_rng()` environment.
`spectral_radius`	Per-lag target spectral radius.
`decay`	Geometric decay rate of spectral radius across lags.

Value

A list of length `lags`, each a p x p matrix.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_genomic_cross_validation

K-fold cross-validation for genomic-prediction accuracy

Description

K-fold cross-validation for genomic-prediction accuracy

Usage

```
morie_genomic_cross_validation(x, y, K = 5, lam = 1, seed = 0)
```

Arguments

x	(n x p) predictor matrix.
y	Numeric response.
K	Number of folds.
lam	Ridge penalty within each fold.
seed	Seed.

Value

list(estimate, r_per_fold, y_hat, mse, mspe, slope, n, K, method).

References

Montesinos Lopez Ch 2.

Examples

```
morie_genomic_cross_validation(x = rnorm(50), y = rnorm(50))
```

morie_geostat_krige

Kriging interpolation via **gstat**

Description

Thin extender over `gstat::krige` for simple, ordinary or universal kriging given a fitted variogram model.

Usage

```
morie_geostat_krige(formula, data, newdata, model, ...)
```

Arguments

`formula`	A formula describing the response and trend terms (e.g. `z ~ 1` for ordinary kriging), as expected by `gstat::krige`.
`data`	Spatial object with the observed locations and response.
`newdata`	Spatial object with the prediction locations.
`model`	A fitted variogram model (e.g. from `gstat::fit.variogram`) describing the spatial covariance.
`...`	Further arguments forwarded to `gstat::krige` (e.g. `nmax`, `nmin`, `block`, `beta`, `debug.level`).

Value

A list with `$method = "gstat::krige"` and `$raw` (a spatial object with kriging predictions and variances).

Examples

```
## Not run:
  if (requireNamespace("gstat", quietly = TRUE) &&
      requireNamespace("sp", quietly = TRUE)) {
    data(meuse, package = "sp")
    data(meuse.grid, package = "sp")
    sp::coordinates(meuse) <- ~ x + y
    sp::coordinates(meuse.grid) <- ~ x + y
    vg <- gstat::variogram(log(zinc) ~ 1, data = meuse)
    mod <- gstat::fit.variogram(vg, gstat::vgm(1, "Sph", 900, 1))
    morie_geostat_krige(log(zinc) ~ 1, meuse, meuse.grid, mod)
  }

## End(Not run)
```

`morie_geostat_variogram`

Empirical variogram via **gstat**

Description

Thin extender over `gstat::variogram` that computes the sample (semi-)variogram of a spatially-indexed response for use in kriging and other geostatistical workflows.

Usage

```
morie_geostat_variogram(formula, data, ...)
```

Arguments

`formula`	A formula describing the response and any trend terms (e.g. `z ~ 1` for an intercept-only model), as expected by `gstat::variogram`.
`data`	A spatial object (e.g. `sp::SpatialPointsDataFrame` or `sf` object) or data frame with coordinates available that `gstat::variogram` can consume.
`...`	Further arguments forwarded to `gstat::variogram` (e.g. `cutoff`, `width`, `cressie`, `cloud`).

Value

A list with `$method = "gstat::variogram"` and `$raw` (a `gstatVariogram` data frame with the binned distances and semivariance estimates).

Examples

```
## Not run:
  if (requireNamespace("gstat", quietly = TRUE) &&
      requireNamespace("sp", quietly = TRUE)) {
    data(meuse, package = "sp")
    sp::coordinates(meuse) <- ~ x + y
    morie_geostat_variogram(log(zinc) ~ 1, data = meuse)
  }

## End(Not run)
```

`morie_ghosal_adaptation` *Adaptive contraction rates over a smoothness grid.*

Description

Adaptive contraction rates over a smoothness grid.

Usage

```
morie_ghosal_adaptation(x, betas = NULL, d = 1)
```

Arguments

`x`	Numeric data vector (used only for sample-size n).
`betas`	Numeric vector of smoothness exponents (default seq(0.5, 3, length.out = 11)).
`d`	Integer dimension (default 1).

Value

Named list with estimate, betas, rates, best_beta, n, d, method.

Examples

```
morie_ghosal_adaptation(x = rnorm(50))
```

`morie_ghosal_bernstein_von_mises` *BvM diagnostic for the mean functional under a DP prior.*

Description

BvM diagnostic for the mean functional under a DP prior.

Usage

```
morie_ghosal_bernstein_von_mises(
  x,
  theta0 = NULL,
  B = 500,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric data vector.
`theta0`	Optional null value for the mean functional.
`B`	Integer number of bootstrap draws (default 500).
`seed`	Integer RNG seed (default 0).
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("ghbvm", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

Named list with estimate, se, theta_hat, z_ks_stat, z_ks_pvalue, wald, wald_pvalue, n, B, method.

Examples

```
morie_ghosal_bernstein_von_mises(x = rnorm(50))
```

`morie_ghosal_contraction_rate`

Minimax posterior-contraction rate

Description

Returns eps_n = n raised to the power of -beta/(2*beta+d).

Usage

```
morie_ghosal_contraction_rate(x, beta = 1, d = 1)
```

Arguments

`x`	Numeric data vector (used only for sample-size n).
`beta`	Numeric smoothness exponent (default 1.0).
`d`	Integer dimension (default 1).

Value

Named list with estimate, log_rate_correction, parametric_rate, n, beta, d, method.

Examples

```
morie_ghosal_contraction_rate(x = rnorm(50))
```

`morie_ghosal_dirichlet_posterior`

Dirichlet-process posterior (conjugate update)

Description

Posterior of G given X_1, ..., X_n for G ~ DP(alpha, G0) with G0 = N(base_mean, base_sd^2). Returns the posterior-mean CDF evaluated on a grid plus the headline `estimate` at `mean(x)`.

Usage

```
morie_ghosal_dirichlet_posterior(
  x,
  alpha = 1,
  base_mean = 0,
  base_sd = 1,
  grid = NULL
)
```

Arguments

`x`	numeric vector.
`alpha`	concentration.
`base_mean`, `base_sd`	base measure (N).
`grid`	optional grid (default: 51 pts spanning x).

Value

named list with `estimate`, `cdf_grid`, `cdf_post`, `cdf_var`, `alpha_post`, `n`, `method`.

Examples

```
morie_ghosal_dirichlet_posterior(x = rnorm(50))
```

`morie_ghosal_dpmixture_density` *DP mixture density estimate (Neal 2000 algorithm 3)*

Description

DP mixture density estimate (Neal 2000 algorithm 3)

Usage

```
morie_ghosal_dpmixture_density(
  x,
  alpha = 1,
  sigma = NULL,
  grid = NULL,
  n_iter = 120,
  burn = 40,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	numeric vector
`alpha`, `sigma`	DP and within-cluster sd (sigma defaults to Silverman bw)
`grid`	evaluation grid
`n_iter`, `burn`, `seed`	Gibbs settings
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("ghdpm", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

named list with `estimate`, `grid`, `density`, `k_post`, `n`

Examples

```
morie_ghosal_dpmixture_density(x = rnorm(50))
```

---

morie_ghosal_empirical_bayes

Empirical-Bayes alpha MLE for a DP, given the observed K_n.

Description

Empirical-Bayes alpha MLE for a DP, given the observed K_n.

Usage

```
morie_ghosal_empirical_bayes(x, alpha_grid = NULL)
```

Arguments

`x`	Numeric data vector.
`alpha_grid`	Optional numeric grid of alpha values to maximise over.

Value

Named list with estimate (alpha-hat), K_n, log_lik_at_estimate, n, method.

Examples

```
morie_ghosal_empirical_bayes(x = rnorm(50))
```

---

morie_ghosal_gp_matern

GP posterior mean with Matern kernel.

Description

GP posterior mean with Matern kernel.

Usage

```
morie_ghosal_gp_matern(
  x,
  y,
  nu = 1.5,
  length_scale = NULL,
  sigma_f = 1,
  noise = NULL,
  x_star = NULL
)
```

Arguments

x	Numeric vector or matrix of input points.
y	Numeric response vector.
nu	Matern smoothness parameter (default 1.5).
length_scale	Optional kernel length-scale.
sigma_f	Numeric signal sd (default 1).
noise	Optional observation noise sd.
x_star	Optional matrix of prediction points (defaults to x).

Value

Named list with estimate, se, mu, sd, length_scale, nu, noise, n, method.

Examples

```
morie_ghosal_gp_matern(x = rnorm(50), y = rnorm(50))
```

---

morie_ghosal_gp_squared_exponential

GP posterior mean with squared-exponential kernel.

Description

GP posterior mean with squared-exponential kernel.

Usage

```
morie_ghosal_gp_squared_exponential(
  x,
  y,
  length_scale = NULL,
  sigma_f = 1,
  noise = NULL,
  x_star = NULL
)
```

Arguments

x	Numeric vector or matrix of input points.
y	Numeric response vector.
length_scale	Optional kernel length-scale.
sigma_f	Numeric signal sd (default 1).
noise	Optional observation noise sd.
x_star	Optional matrix of prediction points (defaults to x).

Value

Named list with estimate, se, mu, sd, length_scale, noise, n, method.

Examples

```
morie_ghosal_gp_squared_exponential(x = rnorm(50), y = rnorm(50))
```

---

morie_ghosal_hierarchical_bayes

Escobar-West augmentation for alpha given K_n with a Gamma(a, b) hyperprior.

Description

Escobar-West augmentation for alpha given K_n with a Gamma(a, b) hyperprior.

Usage

```
morie_ghosal_hierarchical_bayes(
  x,
  a_prior = 1,
  b_prior = 1,
  M = 400,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

x	Numeric data vector.
a_prior	Gamma shape hyperparameter (default 1).
b_prior	Gamma rate hyperparameter (default 1).
M	Integer number of MCMC iterations (default 400).
seed	Integer RNG seed (default 0).
deterministic_seed	Optional integer; if supplied, RNG state is derived via morie_det_rng() keyed on ("ghhbp", deterministic_seed) so Py<->R streams agree on the canonical fixture. When NULL (default) behaviour is unchanged.

Value

Named list with estimate (alpha post mean), alpha_se, alpha_draws, K_n, n, method.

Examples

```
morie_ghosal_hierarchical_bayes(x = rnorm(50))
```

`morie_ghosal_log_density` *Log-spline density estimator (Stone 1990, Ghosal Ch 8).*

Description

Log-spline density estimator (Stone 1990, Ghosal Ch 8).

Usage

```
morie_ghosal_log_density(x, K = 5, grid = NULL)
```

Arguments

`x`	Numeric data vector.
`K`	Integer polynomial degree (default 5).
`grid`	Optional numeric evaluation grid.

Value

Named list with estimate, theta, log_lik, grid, log_density, K, n, method.

Examples

```
morie_ghosal_log_density(x = rnorm(50))
```

`morie_ghosal_moment_matching` *Posterior mean / variance of G(A) for DP(alpha, G0) and A = (A_lower, A_upper].*

Description

Posterior mean / variance of G(A) for DP(alpha, G0) and A = (A_lower, A_upper].

Usage

```
morie_ghosal_moment_matching(
  x,
  alpha = 1,
  A_lower = NULL,
  A_upper = NULL,
  base_mean = 0,
  base_sd = 1
)
```

Arguments

x	Numeric data vector.
alpha	DP concentration parameter (default 1).
A_lower	Optional numeric lower bound of set A (default -Inf).
A_upper	Optional numeric upper bound of set A (default mean(x)).
base_mean	Numeric base-measure mean (default 0).
base_sd	Numeric base-measure sd (default 1).

Value

Named list with estimate, se, prior_mean, prior_var, n_A, n, alpha, method.

Examples

```
morie_ghosal_moment_matching(x = rnorm(50))
```

---

morie_ghosal_neutral_right

Neutral-to-the-right posterior survival (Doksum 1974).

Description

Neutral-to-the-right posterior survival (Doksum 1974).

Usage

```
morie_ghosal_neutral_right(time, event = NULL, c = 1, lam0 = NULL)
```

Arguments

time	Numeric vector of observed times.
event	Optional integer/logical event indicator (1 = event, 0 = censored).
c	Numeric prior concentration (default 1).
lam0	Optional baseline hazard rate.

Value

Named list with estimate, times, S_post, H_post, c, lam0, n, method.

Examples

```
morie_ghosal_neutral_right(time = cumsum(rexp(50)))
```

---

morie_ghosal_np_classification

Probit-GP classifier (Laplace approximation).

Description

Probit-GP classifier (Laplace approximation).

Usage

```
morie_ghosal_np_classification(
  x,
  y,
  length_scale = NULL,
  sigma_f = 1,
  n_iter = 300,
  seed = 0
)
```

Arguments

`x`	Numeric matrix of features.
`y`	Numeric binary labels (0/1).
`length_scale`	Optional kernel length-scale.
`sigma_f`	Numeric signal sd (default 1).
`n_iter`	Integer maximum Laplace iterations (default 300).
`seed`	Integer RNG seed (default 0).

Value

Named list with estimate, p_hat, accuracy, length_scale, n, method.

Examples

```
morie_ghosal_np_classification(x = rnorm(50), y = rnorm(50))
```

morie_ghosal_np_regression

GP nonparametric regression

Description

Wraps `morie_ghosal_gp_squared_exponential`.

Usage

```
morie_ghosal_np_regression(
  x,
  y,
  length_scale = NULL,
  sigma_f = 1,
  noise = NULL
)
```

Arguments

`x`	Numeric vector or matrix of input points.
`y`	Numeric response vector.
`length_scale`	Optional kernel length-scale.
`sigma_f`	Numeric signal sd (default 1).
`noise`	Optional observation noise sd.

Value

Named list with estimate, se, mu, sd, ci_lower, ci_upper, r2, log_marginal, length_scale, noise, n, method.

Examples

```
morie_ghosal_np_regression(x = rnorm(50), y = rnorm(50))
```

morie_ghosal_np_testing

Polya-tree Bayes factor for H0: F = N(loc, scale^2).

Description

Polya-tree Bayes factor for H0: F = N(loc, scale^2).

Usage

```
morie_ghosal_np_testing(x, ref_loc = 0, ref_scale = 1, depth = 6, c = 1)
```

Arguments

`x`	Numeric data vector.
`ref_loc`	Numeric reference location (default 0).
`ref_scale`	Numeric reference scale (default 1).
`depth`	Integer Polya-tree depth (default 6).
`c`	Numeric Polya-tree concentration (default 1).

Value

Named list with statistic (log BF), p_value, BF10, log_BF10, n, depth, method.

Examples

```
morie_ghosal_np_testing(x = rnorm(50))
```

`morie_ghosal_posterior_consistency` *Schwartz posterior-consistency diagnostic (Bayesian bootstrap).*

Description

Schwartz posterior-consistency diagnostic (Bayesian bootstrap).

Usage

```
morie_ghosal_posterior_consistency(
  x,
  ref_loc = NULL,
  ref_scale = NULL,
  eps = 0.1,
  K = 200,
  seed = 0
)
```

Arguments

`x`	Numeric data vector.
`ref_loc`	Optional numeric reference location.
`ref_scale`	Optional numeric reference scale.
`eps`	Numeric KS-distance tolerance (default 0.1).
`K`	Integer number of bootstrap draws (default 200).
`seed`	Integer RNG seed (default 0).

Value

Named list with estimate, ks_mean, ks_se, schwartz_bound, n, eps, method.

Examples

```
morie_ghosal_posterior_consistency(x = rnorm(50))
```

`morie_ghosal_sieve_prior` *Bernstein-polynomial sieve density estimator (Petrone 1999).*

Description

Bernstein-polynomial sieve density estimator (Petrone 1999).

Usage

```
morie_ghosal_sieve_prior(x, K = NULL)
```

Arguments

`x`	Numeric data vector.
`K`	Optional integer sieve degree (default round(n^(1/3))).

Value

Named list with estimate, log_lik_per_obs, weights, K, n, method.

Examples

```
morie_ghosal_sieve_prior(x = rnorm(50))
```

`morie_ghosal_stick_breaking_trunc` *Truncated stick-breaking representation of DP(alpha, G0).*

Description

Truncated stick-breaking representation of DP(alpha, G0).

Usage

```
morie_ghosal_stick_breaking_trunc(
  x,
  alpha = 1,
  K = 50,
  seed = 0,
  base_mean = NULL,
  base_sd = NULL,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric data vector.
`alpha`	DP concentration parameter (default 1).
`K`	Integer truncation level (default 50).
`seed`	Integer RNG seed (default 0).
`base_mean`	Optional base-measure mean.
`base_sd`	Optional base-measure sd.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("ghstk", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

Named list with estimate, weights, atoms, effective_K, trunc_err_bound, n, method.

Examples

```
morie_ghosal_stick_breaking_trunc(x = rnorm(50))
```

`morie_ghosal_survival_beta_process` *Beta-process posterior survival (Hjort 1990).*

Description

Beta-process posterior survival (Hjort 1990).

Usage

```
morie_ghosal_survival_beta_process(time, event = NULL, c = 1, lam0 = NULL)
```

Arguments

`time`	Numeric vector of observed times.
`event`	Optional integer/logical event indicator (1 = event, 0 = censored).
`c`	Numeric prior concentration (default 1).
`lam0`	Optional baseline hazard rate.

Value

Named list with estimate, times, S_post, H_post, c, lam0, n, method.

Examples

```
morie_ghosal_survival_beta_process(time = cumsum(rexp(50)))
```

morie_ghosal_wavelet_prior

Haar-wavelet spike-and-slab BayesThresh estimator (Abramovich 1998).

Description

Haar-wavelet spike-and-slab BayesThresh estimator (Abramovich 1998).

Usage

```
morie_ghosal_wavelet_prior(x, pi = 0.5, sigma = NULL, noise = NULL)
```

Arguments

`x`	Numeric data vector.
`pi`	Numeric prior inclusion probability (default 0.5).
`sigma`	Optional slab sd.
`noise`	Optional noise sd.

Value

Named list with estimate, fitted, noise, sigma, inclusion, n, method.

Examples

```
morie_ghosal_wavelet_prior(x = rnorm(50))
```

morie_gpl_compatible_licenses

Vector of SPDX identifiers recognised as GPL-compatible

Description

Mirrors the FSF list at https://www.gnu.org/licenses/license-list.html. Apache-2.0 is GPL-3 compatible but not GPL-2 compatible; morie is GPL-2.0-only so the choice rests with downstream consumers.

Usage

```
morie_gpl_compatible_licenses()
```

Value

Character vector of SPDX identifiers.

Examples

```
morie_gpl_compatible_licenses()
```

`morie_gradient_boosting_ensemble` *Gradient boosting ensemble (R parity)*

Description

Wraps `gbm::gbm` when available, otherwise falls back to `xgboost` as a portable boosted-trees backend.

Usage

```
morie_gradient_boosting_ensemble(
  x,
  y,
  n_estimators = 100L,
  learning_rate = 0.1,
  max_depth = 3L,
  task = "auto",
  seed = 0L,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric predictor matrix.
`y`	Response.
`n_estimators`	Number of boosting iterations.
`learning_rate`	Shrinkage nu.
`max_depth`	Depth of each tree.
`task`	"auto", "classification", or "regression".
`seed`	RNG seed.
`deterministic_seed`	Integer or NULL. If supplied, the RNG state is derived from the SHA-keyed `morie_det_rng()` so Py<->R streams agree on the canonical fixture. When `NULL` (default), behaviour is unchanged: `seed` drives `set.seed()` directly.

Value

Named list: estimate, train_score, feature_importances, n_estimators, learning_rate, max_depth, task, n, method.

Examples

```
morie_gradient_boosting_ensemble(x = rnorm(50), y = rnorm(50))
```

`morie_gradient_boosting_genomic` *Gradient-boosting genomic predictor (Friedman 2001)*

Description

Uses gbm if available; otherwise base-R boosted stumps.

Usage

```
morie_gradient_boosting_genomic(
  x,
  y,
  markers,
  n_estimators = 100,
  learning_rate = 0.1,
  max_depth = 3,
  seed = 0
)
```

Arguments

`x`	Optional fixed features.
`y`	Numeric response.
`markers`	(n x m) genotype matrix.
`n_estimators`	Boosting rounds.
`learning_rate`	Shrinkage.
`max_depth`	Tree depth (gbm only).
`seed`	Seed.

Value

list(estimate, y_hat, train_loss, se, n, method).

References

Friedman (2001); Montesinos Lopez Ch 9.

Examples

```
morie_gradient_boosting_genomic(
  x = rnorm(50), y = rnorm(50),
  markers = matrix(sample(0:2, 200, TRUE), 50, 4)
)
```

`morie_gradient_descent_vanilla` *Vanilla batch gradient descent for OLS (R parity)*

Description

theta := theta - lr * (2/n) X' (X theta - y), intercept included. Validates against `stats::lm` reference.

Usage

```
morie_gradient_descent_vanilla(x, y, lr = 0.01, n_iter = 1000, tol = 1e-08)
```

Arguments

`x`	Numeric matrix / vector of predictors.
`y`	Numeric response vector.
`lr`	Learning rate.
`n_iter`	Max iterations.
`tol`	L2 step-norm tolerance for early stopping.

Value

Named list with `estimate`, `reference_ols`, `n_iter`, `loss`, `n`, `method`.

Examples

```
morie_gradient_descent_vanilla(x = rnorm(50), y = rnorm(50))
```

`morie_grid_search_cv` *Grid search with cross-validation (R parity)*

Description

Wraps `caret::train` with method = "glm" (classification) or "lm" (regression) by default; users can pass any caret `method`.

Usage

```
morie_grid_search_cv(
  x,
  y,
  method = NULL,
  tune_grid = NULL,
  cv = 5L,
  task = "auto",
  seed = 0L
)
```

Arguments

x	Numeric predictor matrix.
y	Response.
method	caret method id (default chosen by task).
tune_grid	data.frame of hyperparameter combos to evaluate.
cv	CV folds.
task	"auto", "classification", or "regression".
seed	RNG seed.

Value

Named list: estimate (best CV score), best_params, best_score, cv_results_params, cv_results_mean_score, task, n, method.

Examples

```
morie_grid_search_cv(
  x = matrix(rnorm(150), 50, 3), y = rnorm(50),
  method = "lm", tune_grid = data.frame(intercept = c(TRUE, FALSE)),
  cv = 3L, task = "regression", seed = 1L
)
```

---

morie_grm_vanraden *VanRaden Genomic Relationship Matrix*

Description

Computes G = ZZ' / (2 sum p_j(1-p_j)) for method 1 (default), or the per-locus-scaled variant for method 2.

Usage

```
morie_grm_vanraden(markers, method = 1)
```

Arguments

markers	Numeric (n x m) genotype matrix coded (coded 0/1/2).
method	1 or 2 (VanRaden 2008).

Value

Named list with estimate (G matrix), diag_mean, off_mean, p, n, m, method.

References

VanRaden (2008) J Dairy Sci 91:4414. Montesinos Lopez Ch 3.

Examples

```
morie_grm_vanraden(markers = matrix(sample(0:2, 200, TRUE), 50, 4))
```

---

morie_gxe_interaction_model

Two-way GxE ANOVA with EMS variance components

Description

Two-way GxE ANOVA with EMS variance components

Usage

```
morie_gxe_interaction_model(x, y, env)
```

Arguments

x	Genotype IDs (length n).
y	Numeric response.
env	Environment IDs (length n).

Value

list(estimate, g, e, ge, var_g, var_e, var_ge, var_eps, se, n, method).

References

Montesinos Lopez Ch 11.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_hawkes_fit` *Fit a Hawkes (self-exciting point process) model by maximum likelihood*

Description

Fits a one-dimensional Hawkes process with a constant baseline to a vector of event times. The conditional intensity is

$$lambda(t) = nu + eta * sum_{t_j<t} g(t - t_j),$$

where `nu = exp(a0)` is the baseline rate, `eta` the branching ratio, and `g` the chosen triggering kernel.

Usage

```
morie_hawkes_fit(
  times,
  end_time = NULL,
  kernel = c("exponential", "weibull", "lomax", "gamma")
)
```

Arguments

`times`	numeric vector of sorted, non-decreasing event times.
`end_time`	observation horizon; defaults to the last event time.
`kernel`	triggering kernel: one of `"exponential"`, `"weibull"`, `"lomax"`, `"gamma"`.

Details

The negative log-likelihood is evaluated by the shared morie C++ core (the same kernels the Python package uses); without a compiled core it falls back to a pure-R $O(n^2)$ likelihood.

Value

An object of class `morie_hawkes_fit`: a list with the parameter `estimate`, `loglik`, `aic`, `branching_ratio`, `baseline_rate`, `n_events`, `converged` and the `backend` used.

Examples

```
set.seed(1)
ev <- cumsum(rexp(200, rate = 2))
fit <- morie_hawkes_fit(ev, kernel = "exponential")
print(fit)
```

morie_hurst_r *Hurst exponent via rescaled-range (R/S) analysis*

Description

Estimates the Hurst exponent H: a long-memory measure of a time series. $H = 0.5$ indicates uncorrelated (Brownian) increments; $H > 0.5$ indicates persistent (trending) behaviour; $H < 0.5$ indicates anti-persistent (mean-reverting) behaviour.

Usage

```
morie_hurst_r(x)
```

Arguments

`x`	Numeric vector (time series).

Value

List with `H` (numeric) and `interpretation` (`"persistent"`/`"anti-persistent"`/`"random"`).

Examples

```
if (requireNamespace("pracma", quietly = TRUE)) {
  set.seed(1)
  x <- cumsum(rnorm(2048)) # Brownian motion, expected H ~ 0.5
  res <- morie_hurst_r(x)
  res$interpretation
}
```

morie_infer_measurement_level *Infer the measurement level of a vector*

Description

Mirrors the Python `morie.infer_measurement_level()`. Heuristically classifies a vector as one of `"binary"`, `"nominal"`, `"ordinal"`, `"interval"`, or `"ratio"` based on Stevens' (1946) typology.

Usage

```
morie_infer_measurement_level(x)
```

Arguments

x	A vector (any atomic type or factor).

Details

Rules: logical or 2-level factor/character -> "binary"; ordered factor -> "ordinal"; unordered factor or character -> "nominal"; integer/numeric with non-negative range -> "ratio"; otherwise -> "interval".

Value

Character scalar in c("binary", "nominal", "ordinal", "interval", "ratio").

Examples

```
morie_infer_measurement_level(c(0, 1, 1, 0)) # "binary"
morie_infer_measurement_level(factor(c("a", "b", "c"))) # "nominal"
morie_infer_measurement_level(ordered(c("low", "med", "high"))) # "ordinal"
morie_infer_measurement_level(c(1.2, 3.4, 5.6)) # "ratio"
morie_infer_measurement_level(c(-1.5, 0.0, 2.3)) # "interval"
```

---

morie_ingest_bigquery_build_sql

Build a parameter-safe BigQuery SELECT

Description

Builds a SELECT ... FROM `project`.`dataset`.`table` string with identifier validation and backtick-quoting. where is passed through unchanged; callers compose SQL fragments themselves and are responsible for not injecting hostile clauses (same contract as morie_ingest_bigquery_query).

Usage

```
morie_ingest_bigquery_build_sql(
  project,
  dataset,
  table,
  where = NULL,
  limit = NULL,
  select = "*"
)
```

Arguments

`project`, `dataset`, `table`	Fully-qualified BigQuery table reference, e.g. project `"bigquery-public-data"`, dataset `"chicago_crime"`, table `"crime"`.
`where`	Optional raw SQL `WHERE` clause (no leading `WHERE`).
`limit`	Optional `LIMIT`.
`select`	Projection list (default `"*"`).

Value

A SQL string.

Examples

```
morie_ingest_bigquery_build_sql(
  project = "bigquery-public-data",
  dataset = "chicago_crime",
  table   = "crime",
  where   = "year = 2024",
  limit   = 10000L
)
```

`morie_ingest_bigquery_query` *Execute a BigQuery SQL query and return a data.frame*

Description

Runs arbitrary SQL against BigQuery via **bigrquery**, downloads the full result set, and returns it as a base R `data.frame`. Authentication uses Application Default Credentials (the same flow the rest of the HADES-LLM stack uses); to authenticate interactively, run `bigrquery::bq_auth()` first.

Usage

```
morie_ingest_bigquery_query(
  sql,
  billing_project = NULL,
  page_size = 10000L,
  max_rows = Inf,
  quiet = TRUE
)
```

Arguments

`sql`	A SQL string to execute.
`billing_project`	Project to bill the query to. `NULL` falls back to the `GCP_PROJECT` env var, then to ADC.
`page_size`	Rows per download page (forwarded to `bq_table_download`).
`max_rows`	Optional cap on rows downloaded (defaults to `Inf`, i.e. all rows).
`quiet`	Suppress **bigrquery** progress output.

Details

Billing project is resolved from `billing_project`, then the `GCP_PROJECT` environment variable, then ADC discovery; if none of those yields a project the call errors out with a clear message before contacting BigQuery.

Value

A base R `data.frame`.

See Also

`morie_ingest_bigquery_table`, `morie_ingest_bigquery_build_sql`

Examples

```
## Not run:
# Requires the 'bigrquery' package, ADC, and a billing project.
Sys.setenv(GCP_PROJECT = "my-billing-project")
df <- morie_ingest_bigquery_query(
  "SELECT year, COUNT(*) AS n
     FROM `bigquery-public-data.chicago_crime.crime`
    GROUP BY year
    ORDER BY year"
)
head(df)

## End(Not run)
```

`morie_ingest_bigquery_table` *Pull a BigQuery table (or filtered slice) into a data.frame*

Description

Convenience wrapper around `morie_ingest_bigquery_build_sql` + `morie_ingest_bigquery_query`: builds a validated, backtick-quoted `SELECT` against a fully-qualified table and downloads the result.

Usage

```
morie_ingest_bigquery_table(
  project,
  dataset,
  table,
  where = NULL,
  limit = NULL,
  select = "*",
  billing_project = NULL,
  page_size = 10000L,
  max_rows = Inf,
  quiet = TRUE
)
```

Arguments

`project`, `dataset`, `table`	Fully-qualified BigQuery table reference, e.g. `project = "bigquery-public-data"`, `dataset = "chicago_crime"`, `table = "crime"`.
`where`	Optional raw SQL `WHERE` clause.
`limit`	Optional `LIMIT`.
`select`	Projection list (default `"*"`).
`billing_project`	Billing project; falls back to `GCP_PROJECT`, then ADC.
`page_size`	Rows per download page.
`max_rows`	Optional cap on rows downloaded.
`quiet`	Suppress **bigrquery** progress output.

Value

A base R `data.frame`.

See Also

`morie_ingest_bigquery_query`

Examples

```
## Not run:
# Requires the 'bigrquery' package, ADC, and a billing project.
df <- morie_ingest_bigquery_table(
  project = "bigquery-public-data",
  dataset = "chicago_crime",
  table   = "crime",
  where   = "year = 2024",
  limit   = 10000L,
  billing_project = "my-billing-project"
)
```

```
head(df)

## End(Not run)
```

`morie_ingest_chicago_crime`
Pull the City of Chicago "Crimes - 2001 to Present" feed

Description

Returns a data.frame with the documented Socrata schema (snake_case column names preserved): `id`, `case_number`, `date`, `block`, `iucr`, `primary_type`, `description`, `location_description`, `arrest`, `domestic`, `beat`, `district`, `ward`, `community_area`, `fbi_code`, `x_coordinate`, `y_coordinate`, `year`, `updated_on`, `latitude`, `longitude`.

Usage

```
morie_ingest_chicago_crime(
  year = NULL,
  where = NULL,
  max_features = NULL,
  app_token = NULL,
  user_agent = .MORIE_CHICAGO_DEFAULT_UA,
  timeout = .MORIE_CHICAGO_DEFAULT_TIMEOUT
)
```

Arguments

`year`	Optional reporting year (e.g. 2024); when set, applies the server-side SoQL filter `year = <year>`.
`where`	Optional raw SoQL `$where` clause (overrides `year`).
`max_features`	Optional hard cap on returned rows.
`app_token`	Optional Socrata X-App-Token for higher rate limits.
`user_agent`, `timeout`	Standard request knobs.

Value

A base R `data.frame`.

See Also

`morie_ingest_chicago_socrata`, `morie_ingest_bigquery_table` for the BigQuery public-data mirror (`bigquery-public-data.chicago_crime`).

Examples

```
## Not run:
df <- morie_ingest_chicago_crime(year = 2024, max_features = 10000L)
head(df)

## End(Not run)
```

`morie_ingest_chicago_crime_bigquery`
BigQuery mirror of the Chicago crime feed

Description

Convenience wrapper around `morie_ingest_bigquery_table` that pulls `bigquery-public-data.chicago_crime.` - the Google BigQuery public-data mirror of the Socrata feed served by `morie_ingest_chicago_crime`. Use this path when you want SQL-side filtering or the full historical depth of the dataset without paging through SoQL.

Usage

```
morie_ingest_chicago_crime_bigquery(
  where = NULL,
  year = NULL,
  limit = NULL,
  select = "*",
  billing_project = NULL,
  page_size = 10000L,
  max_rows = Inf,
  quiet = TRUE
)
```

Arguments

`where`	Optional raw SQL `WHERE` clause (no leading `WHERE`).
`year`	Convenience shortcut: when `where` is `NULL` and `year` is set, applies `year = <year>`.
`limit`	Optional `LIMIT`.
`select`	Projection list (default `"*"`).
`billing_project`, `page_size`, `max_rows`, `quiet`	Forwarded to `morie_ingest_bigquery_table`.

Details

Requires the optional **bigrquery** package and a billing project (`billing_project` arg or `GCP_PROJECT` env var); public datasets are billed to the caller's project, not the dataset owner's.

Value

A base R `data.frame`.

See Also

`morie_ingest_chicago_crime`, `morie_ingest_bigquery_table`

---

morie_ingest_chicago_resources

Built-in Chicago / Socrata resource registry

Description

Returns the canonical Chicago open-data Socrata endpoints morie ships with as a flat data.frame. Useful for discovery and for the CLI `--list` surface.

Usage

```
morie_ingest_chicago_resources()
```

Value

A base R `data.frame` with columns `name`, `url`.

---

morie_ingest_chicago_socrata

Fetch every row from a Socrata SoDA JSON endpoint

Description

Pages transparently through `$offset` until either the server returns fewer rows than `page_size` (the last page) or `max_features` is reached. Works against any Socrata-shaped portal (Chicago, NYC, Seattle, etc.).

Usage

```
morie_ingest_chicago_socrata(
  resource_url,
  where = NULL,
  select = NULL,
  order = NULL,
  page_size = .MORIE_CHICAGO_DEFAULT_PAGE_SIZE,
  max_features = NULL,
  app_token = NULL,
  user_agent = .MORIE_CHICAGO_DEFAULT_UA,
  timeout = .MORIE_CHICAGO_DEFAULT_TIMEOUT
)
```

Arguments

`resource_url`	Full Socrata resource URL ending in `.json`, e.g. `"https://data.cityofchicago.org/resource`
`where`, `select`, `order`	SoQL clauses; see https://dev.socrata.com/docs/queries/. `where` is the equivalent of SQL `WHERE`.
`page_size`	Rows per request (capped at 50,000 server-side).
`max_features`	Optional hard cap on total returned rows.
`app_token`	Optional Socrata application token (anonymous calls share a throttled pool; tokens give per-app quotas).
`user_agent`, `timeout`	Standard request knobs.

Value

A base R `data.frame`.

Examples

```
## Not run:
df <- morie_ingest_chicago_socrata(
  "https://data.cityofnewyork.us/resource/uip8-fykc.json",
  where = "arrest_year = 2023",
  max_features = 5000L
)

## End(Not run)
```

`morie_ingest_cihi_xlsx` *Download a CIHI indicator .xlsx data table*

Description

Download a CIHI indicator .xlsx data table

Usage

```
morie_ingest_cihi_xlsx(
  url,
  sheet = NULL,
  timeout = 120,
  user_agent = "morie/r (+https://hadesllm.com)",
  ...
)
```

Arguments

`url`	Direct URL of the CIHI .xlsx data table.
`sheet`	Worksheet name or 1-based index. NULL = largest sheet.
`timeout`	HTTP timeout in seconds (default 120).
`user_agent`	User-Agent string.
`...`	forwarded to readxl::read_excel.

Value

base R data.frame.

`morie_ingest_ckan_fetch_package_csvs` *Fetch every CSV / TSV resource in a CKAN package*

Description

Mirrors the Python `fetch_package_csvs` helper: pulls the package metadata, walks its `resources` list, and downloads each CSV / TSV into a named list of data.frames keyed by resource `name` (falling back to `url`, then `id`). Non-CSV / TSV resources are skipped; individual download failures are captured as a single-row error data.frame keyed `_failed_<name>` so the overall fetch still returns the successful ones.

Usage

```
morie_ingest_ckan_fetch_package_csvs(
  portal,
  package_id,
  api_key = NULL,
  user_agent = .MORIE_CKAN_DEFAULT_UA,
  timeout = .MORIE_CKAN_DEFAULT_TIMEOUT
)
```

Arguments

`portal`	Base URL of the CKAN portal.
`package_id`	Package id or slug.
`api_key`	Optional CKAN API key.
`user_agent`	User-Agent header sent with the request.
`timeout`	HTTP timeout in seconds.

Value

A named list of data.frames.

`morie_ingest_ckan_package_search` *Search a CKAN portal for packages (raw)*

Description

Calls the CKAN `package_search` Action verb against a portal base URL and returns the raw result list (`count`, `results`, etc.). For a flattened metadata data.frame, use `morie_ingest_ckan_search_packages`.

Usage

```
morie_ingest_ckan_package_search(
  portal,
  query = NULL,
  fq = NULL,
  rows = 100L,
  start = 0L,
  api_key = NULL,
  user_agent = .MORIE_CKAN_DEFAULT_UA,
  timeout = .MORIE_CKAN_DEFAULT_TIMEOUT
)
```

Arguments

`portal`	Base URL of the CKAN portal, e.g. `"https://open.canada.ca/data"`. Trailing slash optional.
`query`	Free-text `q=` query, or `NULL`.
`fq`	CKAN `fq=` filter query (Solr-style), or `NULL`.
`rows`	Maximum rows to return (default 100).
`start`	Pagination offset (default 0).
`api_key`	Optional CKAN API key (rare for open portals).
`user_agent`	User-Agent header sent with the request.
`timeout`	HTTP timeout in seconds.

Value

A named list as returned by the CKAN Action API.

Examples

```
## Not run:
res <- morie_ingest_ckan_package_search(
  "https://open.canada.ca/data",
  query = "corrections"
)
```

```
length(res$results)

## End(Not run)
```

`morie_ingest_ckan_package_show`

Fetch one CKAN package's metadata

Description

Calls the CKAN `package_show` verb. The returned list contains `title`, `notes`, `resources`, etc. - resources are the individual downloadable files in the package.

Usage

```
morie_ingest_ckan_package_show(
  portal,
  package_id,
  api_key = NULL,
  user_agent = .MORIE_CKAN_DEFAULT_UA,
  timeout = .MORIE_CKAN_DEFAULT_TIMEOUT
)
```

Arguments

`portal`	Base URL of the CKAN portal.
`package_id`	Package id or slug, e.g. `"canadian-postsecondary-alcohol-and-drug-use-survey"`.
`api_key`	Optional CKAN API key.
`user_agent`	User-Agent header sent with the request.
`timeout`	HTTP timeout in seconds.

Value

The package metadata list.

`morie_ingest_ckan_read_resource`

Download a CKAN resource as a data.frame

Description

`url_or_id` may be a direct download URL (as it appears in `resource$url`) or a CKAN resource id, in which case the URL is resolved via `morie_ingest_ckan_resource_show`.

Usage

```
morie_ingest_ckan_read_resource(
  portal,
  url_or_id,
  as_format = NULL,
  api_key = NULL,
  user_agent = .MORIE_CKAN_DEFAULT_UA,
  timeout = .MORIE_CKAN_DEFAULT_TIMEOUT
)
```

Arguments

`portal`	Base URL of the CKAN portal (only used when `url_or_id` is a bare resource id).
`url_or_id`	A direct URL or a CKAN resource id.
`as_format`	Optional format override (`"csv"`, `"tsv"`, `"xlsx"`, `"json"`, `"parquet"`).
`api_key`	Optional CKAN API key.
`user_agent`	User-Agent header sent with the request.
`timeout`	HTTP timeout in seconds.

Details

Format detection: if `as_format` is given it wins. Otherwise the extension is sniffed off the URL; unknown extensions fall back to CSV (matching the Python helper).

Excel / JSON / Parquet readers require optional dependencies (**readxl** / **jsonlite** / **arrow**) and error with an install hint if missing.

Value

A base R `data.frame`.

`morie_ingest_ckan_resource_show` *Fetch one CKAN resource's metadata*

Description

Calls the CKAN `resource_show` verb to resolve a resource id into its download URL plus metadata.

Usage

```
morie_ingest_ckan_resource_show(
  portal,
  resource_id,
  api_key = NULL,
  user_agent = .MORIE_CKAN_DEFAULT_UA,
  timeout = .MORIE_CKAN_DEFAULT_TIMEOUT
)
```

Arguments

`portal`	Base URL of the CKAN portal.
`resource_id`	CKAN resource id (UUID).
`api_key`	Optional CKAN API key.
`user_agent`	User-Agent header sent with the request.
`timeout`	HTTP timeout in seconds.

Value

The resource metadata list.

`morie_ingest_ckan_search_packages`

Search a CKAN portal and return a flat metadata data.frame

Description

Convenience wrapper over `morie_ingest_ckan_package_search` that flattens the most-useful columns into a single data.frame: `id`, `name`, `title`, `organization`, `license_id`, `metadata_modified`, `num_resources`, `url` (the canonical `<portal>/dataset/<name>` URL).

Usage

```
morie_ingest_ckan_search_packages(
  portal,
  query,
  rows = 50L,
  api_key = NULL,
  user_agent = .MORIE_CKAN_DEFAULT_UA,
  timeout = .MORIE_CKAN_DEFAULT_TIMEOUT
)
```

Arguments

portal	Base URL of the CKAN portal.
query	Free-text query string.
rows	Maximum rows to return (default 50).
api_key	Optional CKAN API key.
user_agent	User-Agent header sent with the request.
timeout	HTTP timeout in seconds.

Value

A base R data.frame.

---

morie_ingest_forensics_namus_missing

Pull NamUs missing-persons case metadata

Description

Posts a JSON-body search request to the NamUs MissingPersons endpoint (/api/CaseSets/NamUs/MissingPerson and pages through the results. Returns morie's documented schema - case_number, state, county, dlc_date (date last contact), sex, race, age_min, age_max, height_cm_min, height_cm_max, weight_kg_min, weight_kg_max, first_name, last_name, city, circumstances.

Usage

```
morie_ingest_forensics_namus_missing(
  state = NULL,
  max_features = NULL,
  page_size = 200L,
  user_agent = .MORIE_FORENSICS_DEFAULT_UA,
  timeout = .MORIE_FORENSICS_DEFAULT_TIMEOUT
)
```

Arguments

state	Two-letter US state code; NULL returns the national feed.
max_features	Optional hard cap on returned rows.
page_size	Records per request (default 200).
user_agent, timeout	Standard request knobs.

Value

A base R data.frame.

Examples

```
## Not run:
df <- morie_ingest_forensics_namus_missing(state = "CA",
                                           max_features = 1000L)
head(df)

## End(Not run)
```

`morie_ingest_forensics_nibrs`
Pull FBI NIBRS offence-event records via Crime Data Explorer

Description

Queries the FBI Crime Data Explorer NIBRS endpoint (`/crime/fbi/cde/nibrs/<state>/<offense>?year=...`). Requires an API key from https://api.data.gov/signup/; pass via `api_key=` or set the `FBI_CDE_API_KEY` environment variable. Returns one row per offence-event with nested sub-objects flattened using dotted keys (`offense.code`, `victim.age`, ...).

Usage

```
morie_ingest_forensics_nibrs(
  year,
  offense = NULL,
  state = NULL,
  api_key = NULL,
  max_features = NULL,
  page_size = 500L,
  timeout = .MORIE_FORENSICS_DEFAULT_TIMEOUT
)
```

Arguments

`year`	Reporting year (e.g. 2023). Required - CDE forces a year scope.
`offense`	NIBRS offence slug (e.g. `"aggravated-assault"`, `"burglary"`); `NULL` returns all offences.
`state`	Two-letter US state code (e.g. `"GA"`). `NULL` returns the national feed (very large; use `max_features`).
`api_key`	FBI CDE API key; falls back to `$FBI_CDE_API_KEY`.
`max_features`	Optional hard cap on returned rows.
`page_size`	CDE page size; server-side cap varies by endpoint.
`timeout`	HTTP timeout in seconds.

Value

A base R `data.frame`, one row per offence-event.

Examples

```
## Not run:
df <- morie_ingest_forensics_nibrs(
  year = 2023, offense = "aggravated-assault", state = "GA",
  api_key = Sys.getenv("FBI_CDE_API_KEY"),
  max_features = 5000L
)
head(df)

## End(Not run)
```

---

morie_ingest_forensics_nist_rds

Pull NIST Reference Datasets (RDS) catalog metadata

Description

The raw reference datasets (CSAFE bullets/cartridges, NSRL hash library, ...) are multi-gigabyte and shipped on dedicated download servers; this function returns only the catalog records so the caller can pick what to download separately.

Usage

```
morie_ingest_forensics_nist_rds(
  dataset_id = NULL,
  query = NULL,
  max_features = NULL,
  page_size = 50L,
  raw = FALSE,
  timeout = .MORIE_FORENSICS_DEFAULT_TIMEOUT
)
```

Arguments

`dataset_id`	Specific NIST RDS / EDI id (e.g. `"ark:/88434/mds2-2418"`). When set, returns a single-row frame.
`query`	Free-text search over title / description / keyword. Ignored when `dataset_id` is set.
`max_features`	Optional hard cap on returned rows.
`page_size`	Records per request (default 50).
`raw`	If `TRUE`, return the raw catalog JSON columns instead of morie's flattened schema.
`timeout`	HTTP timeout in seconds.

Value

A base R `data.frame`.

`morie_ingest_statcan_cansim` *Fetch a Statistics Canada NDM / cansim table*

Description

Convenience wrapper around the CRAN **cansim** package, which talks to the Statistics Canada NDM ("cansim") tabular data API. Use this for canonical CANSIM tables (e.g. `"35-10-0177-01"`) rather than for PUMF `_CSV.zip` downloads — those go through `morie_ingest_statcan_csv`.

Usage

```
morie_ingest_statcan_cansim(
  table_id,
  language = c("eng", "fra"),
  refresh = FALSE,
  ...
)
```

Arguments

`table_id`	A StatCan / NDM table identifier, e.g. `"35-10-0177"` or `"35-10-0177-01"`.
`language`	One of `"eng"` or `"fra"`.
`refresh`	If `TRUE`, force **cansim** to re-download rather than using its on-disk cache.
`...`	Further arguments forwarded to `get_cansim`.

Details

If the `STATCAN_API_KEY` environment variable is set, it is passed to `cansim::set_cansim_api_key()` so authenticated rate limits apply.

Value

A base R `data.frame`.

See Also

`morie_ingest_statcan_csv`

Examples

```
## Not run:
# Requires the 'cansim' package and network access.
df <- morie_ingest_statcan_cansim("35-10-0177")
head(df)

## End(Not run)
```

`morie_ingest_statcan_csv` *Download a StatCan PUMF / CSV product*

Description

Downloads a Statistics Canada `_CSV.zip` product from `www150.statcan.gc.ca`, extracts a CSV member, and returns the contents as a base R `data.frame`. The archive is streamed to a session-scoped tempfile (PUMF zips can be hundreds of megabytes), and the tempfile is removed when the function returns. Nothing is written under `~/.cache` unless the caller explicitly opts in via `morie_cache_dir`.

Usage

```
morie_ingest_statcan_csv(
  url,
  member = NULL,
  timeout = 600,
  user_agent = "morie/r (+https://hadesllm.com)",
  ...
)
```

Arguments

`url`	Direct URL of the StatCan `.zip` product, e.g. `https://www150.statcan.gc.ca/n1/pub/82m001`
`member`	Name of the CSV inside the archive; defaults to the first `.csv` entry.
`timeout`	HTTP timeout in seconds (default 600).
`user_agent`	User-Agent string sent with the request.
`...`	Further arguments forwarded to `read_csv` (or `read.csv` if **readr** is unavailable).

Details

Note that a StatCan *catalogue* page (e.g. `/n1/en/catalogue/82M0013X`) is only an HTML index — the actual data is linked from the *product* page (`/n1/pub/82m0013x/82m0013x2024001-eng.htm`), which points at the real `..._CSV.zip`.

Value

A base R `data.frame`.

See Also

`morie_ingest_statcan_cansim`, `morie_cache_dir`

Examples

```
## Not run:
# Requires network access.
url <- paste0(
  "https://www150.statcan.gc.ca/n1/pub/82m0013x/",
  "2024001/2022_CSV.zip"
)
df <- morie_ingest_statcan_csv(url)
head(df)

## End(Not run)
```

`morie_ingest_tps_feature_layer` *Fetch every feature from a TPS ArcGIS FeatureServer layer*

Description

ArcGIS FeatureServer queries cap at the layer's server-side `maxRecordCount` (2,000 for the TPS layers). This function pages through transparently using `resultOffset` and the `exceededTransferLimit` flag, emitting one data.frame.

Usage

```
morie_ingest_tps_feature_layer(
  layer_url,
  where = "1=1",
  out_fields = "*",
  return_geometry = FALSE,
  page_size = 2000L,
  max_features = NULL,
  user_agent = .MORIE_TPS_DEFAULT_UA,
  timeout = .MORIE_TPS_DEFAULT_TIMEOUT
)
```

Arguments

`layer_url`	Full URL to a FeatureServer layer, e.g. one of the entries in `morie_ingest_tps_layers`.
`where`	ArcGIS WHERE clause. Default `"1=1"` fetches everything. Examples: `"OCC_YEAR = 2024"`, `"OCC_YEAR BETWEEN 2020 AND 2025"`.
`out_fields`	Comma-separated attribute list, or `"*"`.
`return_geometry`	If `TRUE`, includes `geom_x` / `geom_y` columns (longitude / latitude in EPSG:4326).
`page_size`	Records per request; clamped server-side to 2,000 for TPS layers.
`max_features`	Optional hard cap on total returned rows.
`user_agent`, `timeout`	Standard request knobs.

Value

A base R `data.frame`.

Examples

```
## Not run:
df <- morie_ingest_tps_feature_layer(
  morie_ingest_tps_layers()$url[
    morie_ingest_tps_layers()$name == "major-crime"
  ],
  where = "OCC_YEAR >= 2023",
  max_features = 5000L
)
nrow(df)

## End(Not run)
```

---

morie_ingest_tps_fetch

Fetch a TPS open-data layer by short name

Description

Convenience wrapper around `morie_ingest_tps_feature_layer` that takes a registry short-name (e.g. `"major-crime"`, `"shooting-firearms"`) instead of a raw FeatureServer URL.

Usage

```
morie_ingest_tps_fetch(
  layer,
  year = NULL,
  where = NULL,
  return_geometry = FALSE,
  max_features = NULL,
  ...
)
```

Arguments

`layer`	Short name from `morie_ingest_tps_layers`.
`year`	Optional shortcut for `OCC_YEAR = <year>` when `where` is `NULL`.
`where`	Raw ArcGIS WHERE clause (overrides `year`).
`return_geometry`	Include longitude/latitude columns.
`max_features`	Optional hard cap on rows.
`...`	Forwarded to `morie_ingest_tps_feature_layer`.

Value

A base R `data.frame`.

`morie_ingest_tps_layers` *Built-in TPS open-data layer registry*

Description

Returns the canonical Toronto Police Service open-data ArcGIS FeatureServer layer URLs morie ships with as a flat data.frame. Useful for discovery and for the CLI `--list` surface.

Usage

```
morie_ingest_tps_layers()
```

Value

A base R `data.frame` with columns `name`, `url`.

Examples

```
morie_ingest_tps_layers()
```

`morie_inspect_output` *Inspect a serialised analysis output (JSON, CSV, or RDS)*

Description

Mirrors the Python `morie.inspect_output()`. Reads a structured output file and returns a brief summary of its contents.

Usage

```
morie_inspect_output(path)
```

Arguments

`path`	Path to a JSON, CSV, or RDS file.

Details

Supported formats: `.json` (via `jsonlite`), `.csv` (via base `utils::read.csv`), `.rds` (via `base::readRDS`).

Value

A list with components `path`, `format`, `exists`, `size_bytes`, and (on success) `contents_preview` plus type-appropriate metadata.

Examples

```
tmp <- tempfile(fileext = ".json")
if (requireNamespace("jsonlite", quietly = TRUE)) {
  jsonlite::write_json(list(estimate = 0.123, se = 0.045), tmp)
  morie_inspect_output(tmp)
  unlink(tmp)
}
```

`morie_install_extras` *Install morie's optional dependencies (interactive helper)*

Description

morie's `Suggests:` list spans ~50 R packages (causal/ML/spatial/IO families). CRAN policy requires us to leave their install to the user (no `install.packages()` at load time, no user-home writes). This helper resolves which Suggests are missing and (with user confirmation) installs them, plus prints platform-specific install hints for the system libraries morie's C/C++ backends use (`libcurl`, `libsodium`, optional `liboqs`).

Usage

```
morie_install_extras(
  which = "missing",
  ask = interactive(),
  repos = NULL,
  dependencies = NA,
  ...
)
```

Arguments

`which`	Either `"missing"` (install only missing Suggests – the default), `"all"` (install/upgrade every Suggests), or a character vector of specific package names.
`ask`	Logical. If `TRUE` (default), prompt the user before installing. Set to `FALSE` for non-interactive CI workflows.
`repos`	The CRAN-like repository URL(s) to install from. Default uses `getOption("repos")`, falling back to the RStudio CRAN mirror.
`dependencies`	Passed through to `utils::install.packages()`. Default `NA` honours the package's `Depends`/`Imports`/`LinkingTo` only – not the optional Suggests-of-Suggests cascade.
`...`	Extra args forwarded to `utils::install.packages()`.

Value

Invisibly: a list with `installed` (character of packages added this call), `already_present` (already installed), `failed` (failed to install), and `system_libs` (named logical of detected system libraries).

System libraries

morie's compiled backends need three C libraries available at build time. Two are typically pre-installed on developer machines; one is optional and gates the post-quantum cryptography family. Install BEFORE installing/upgrading morie so the configure-time probes pick them up.

- **libcurl** (required for HTTP fetchers)
  - Debian/Ubuntu: `sudo apt-get install libcurl4-openssl-dev`
  - Fedora/RHEL: `sudo dnf install libcurl-devel`
  - macOS: pre-installed (Apple's libcurl); or `brew install curl`
  - Windows: bundled with Rtools
- **libsodium** (required for ChaCha20-Poly1305 + HKDF-SHA256)
  - Debian/Ubuntu: `sudo apt-get install libsodium-dev`
  - Fedora/RHEL: `sudo dnf install libsodium-devel`
  - macOS: `brew install libsodium`
- **liboqs** (optional, gates ML-KEM-768 + ML-DSA-65)
  - Debian/Ubuntu: build from source (not yet packaged); https://github.com/open-quantum-safe/liboqs
  - Fedora/RHEL: `sudo dnf install liboqs-devel` (recent releases)
  - macOS: `brew install liboqs`

Examples

```
## Not run:
  # Interactive: install whichever Suggests are missing
  morie_install_extras()

  # CI / scripted: install all, no prompt
  morie_install_extras(which = "all", ask = FALSE)

  # Just one family
  morie_install_extras(which = c("hawkes", "sf", "spdep"))

## End(Not run)
```

`morie_is_over_legal_limit` *Test whether an eBAC exceeds a legal driving limit*

Description

Test whether an eBAC exceeds a legal driving limit

Usage

```
morie_is_over_legal_limit(ebac, limit = 0.08)
```

Arguments

`ebac`	Numeric eBAC value (e.g. from `morie_calculate_ebac()`).
`limit`	Legal threshold (default 0.08, the per-se limit in most Canadian and US jurisdictions).

Value

Integer 1 if `ebac >= limit`, 0 otherwise. (Integer, not logical, to match the Python sibling and ease binary-outcome modelling.)

Examples

```
morie_is_over_legal_limit(0.09)
morie_is_over_legal_limit(0.05, limit = 0.05)
```

`morie_iv_anderson_rubin` *Anderson-Rubin (AR) weak-IV-robust test*

Description

Anderson-Rubin (AR) weak-IV-robust test

Usage

```
morie_iv_anderson_rubin(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  beta0 = NULL,
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.
`beta0`	Numeric scalar or vector; the structural-coefficient value(s) to test under H0. Length must match `length(endogenous)`.
`alpha`	Significance level (default `0.05`); controls the confidence-set / acceptance-region cut-off.

`morie_iv_anderson_rubin_ci` *Grid-based Anderson-Rubin confidence interval for a single endogenous variable.*

Description

Grid-based Anderson-Rubin confidence interval for a single endogenous variable.

Usage

```
morie_iv_anderson_rubin_ci(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  grid_min = -10,
  grid_max = 10,
  grid_n = 200,
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.

exogenous	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. NULL (default) for a just-identified design.
grid_min	Numeric; lower bound of the AR confidence-set grid search over candidate beta0 values.
grid_max	Numeric; upper bound of the AR confidence-set grid.
grid_n	Integer; number of grid points used in morie_iv_anderson_rubin_ci (default 100).
alpha	Significance level (default 0.05); controls the confidence-set / acceptance-region cut-off.

---

morie_iv_conditional_lr

Conditional likelihood-ratio (CLR) test of Moreira (2003)

Description

Conditional likelihood-ratio (CLR) test of Moreira (2003)

Usage

```
morie_iv_conditional_lr(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  beta0 = 0
)
```

Arguments

data	A data.frame (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
outcome	Character; column name of the response variable.
endogenous	Character vector; column names of the endogenous regressors.
instruments	Character vector; column names of the instrumental variables.
exogenous	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. NULL (default) for a just-identified design.
beta0	Numeric scalar or vector; the structural-coefficient value(s) to test under H0. Length must match length(endogenous).

morie_iv_control_function

Control-function (residual augmentation) IV

Description

Control-function (residual augmentation) IV

Usage

```
morie_iv_control_function(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  robust = TRUE,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`robust`	Logical; if `TRUE` use HC1 robust standard errors.
`alpha`	Significance level for confidence intervals.

morie_iv_cragg_donald

Cragg-Donald weak-instrument F statistic

Description

Computes the Cragg-Donald (1993) weak-instrument statistic. The statistic is a function of the first-stage regression and is independent of the outcome variable; `outcome` only needs to name a numeric column in `data` so **ivreg** can compile a formula. When `outcome = NULL` (default), the first endogenous regressor is reused as the outcome – works because **ivreg**'s weak-IV diagnostic comes from the first stage regardless of `y`.

Usage

```
morie_iv_cragg_donald(
  data,
  endogenous,
  instruments,
  exogenous = NULL,
  outcome = NULL
)
```

Arguments

`data`	Data frame.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional exogenous covariates.
`outcome`	Optional outcome column name. Default `NULL` reuses `endogenous[1]`; the resulting F-statistic is unaffected because Cragg-Donald only reads the first stage.

Value

Named list with `statistic`, `p_value`, `name`, `details`.

`morie_iv_cue_gmm` *Continuously-Updated GMM (CUE-GMM)*

Description

Continuously-Updated GMM (CUE-GMM)

Usage

```
morie_iv_cue_gmm(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  max_iter = 100,
  tol = 1e-08,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`max_iter`	Outer iteration cap (default 100).
`tol`	Convergence tolerance on the objective.
`alpha`	Significance level for confidence intervals.

`morie_iv_diagnostics` *Composite IV diagnostics*

Description

Composite IV diagnostics

Usage

```
morie_iv_diagnostics(data, outcome, endogenous, instruments, exogenous = NULL)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.

morie_iv_durbin_wu_hausman

Durbin-Wu-Hausman test of endogeneity

Description

Durbin-Wu-Hausman test of endogeneity

Usage

```
morie_iv_durbin_wu_hausman(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL
)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.

morie_iv_first_stage_diagnostics

First-stage F-statistics and partial R^2

Description

First-stage F-statistics and partial R^2

Usage

```
morie_iv_first_stage_diagnostics(
  data,
  endogenous,
  instruments,
  exogenous = NULL
)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.

`morie_iv_gmm` *Generalised Method of Moments (GMM) IV*

Description

Two-step efficient GMM via `gmm::gmm`; falls back to 2SLS otherwise.

Usage

```
morie_iv_gmm(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  weight_matrix = "optimal",
  robust = TRUE,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`weight_matrix`	One of `"optimal"` (default, two-step) or `"identity"` (one-step / 2SLS-equivalent).
`robust`	Logical; if `TRUE` use HC1 robust standard errors.
`alpha`	Significance level for confidence intervals.

morie_iv_hansen_j *Hansen J test of overidentifying restrictions (robust)*

Description

Hansen J test of overidentifying restrictions (robust)

Usage

```
morie_iv_hansen_j(data, outcome, endogenous, instruments, exogenous = NULL)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.

morie_iv_hausman *Hausman test: OLS vs 2SLS*

Description

Hausman test: OLS vs 2SLS

Usage

```
morie_iv_hausman(data, outcome, endogenous, instruments, exogenous = NULL)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.

morie_iv_jive *Jackknife IV (JIVE; Angrist, Imbens & Krueger 1999)*

Description

Jackknife IV (JIVE; Angrist, Imbens & Krueger 1999)

Usage

```
morie_iv_jive(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  alpha = 0.05
)
```

Arguments

data	Data frame.
outcome	Name of the outcome column.
endogenous	Character vector of endogenous regressor names.
instruments	Character vector of excluded-instrument names.
exogenous	Optional character vector of exogenous covariate names.
alpha	Significance level for confidence intervals.

morie_iv_kleibergen_paap *Kleibergen-Paap rank statistic*

Description

Kleibergen-Paap rank statistic

Usage

```
morie_iv_kleibergen_paap(data, endogenous, instruments, exogenous = NULL)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.
`exogenous`	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. `NULL` (default) for a just-identified design.

`morie_iv_liml` *Limited-Information Maximum Likelihood (LIML)*

Description

Solves the LIML eigenvalue problem; falls back to `ivreg::ivreg(...,method = "M")` if available.

Usage

```
morie_iv_liml(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  robust = TRUE,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`robust`	Logical; if `TRUE` use HC1 robust standard errors.
`alpha`	Significance level for confidence intervals.

morie_iv_panel — *Panel IV with unit (and optional time) fixed effects via within-transform*

Description

Panel IV with unit (and optional time) fixed effects via within-transform

Usage

```
morie_iv_panel(
  data,
  outcome,
  endogenous,
  instruments,
  unit,
  exogenous = NULL,
  time_fe = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`unit`	Cluster / unit identifier column.
`exogenous`	Optional character vector of exogenous covariate names.
`time_fe`	Optional time-FE column.
`alpha`	Significance level for confidence intervals.

morie_iv_probit — *IV Probit (Rivers-Vuong control function)*

Description

IV Probit (Rivers-Vuong control function)

Usage

```
morie_iv_probit(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`alpha`	Significance level for confidence intervals.

`morie_iv_residual_analysis` *IV residual analysis*

Description

IV residual analysis

Usage

```
morie_iv_residual_analysis(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL
)
```

Arguments

`data`	A `data.frame` (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
`outcome`	Character; column name of the response variable.
`endogenous`	Character vector; column names of the endogenous regressors.
`instruments`	Character vector; column names of the instrumental variables.

exogenous	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. NULL (default) for a just-identified design.

---

morie_iv_sargan *Sargan test of overidentifying restrictions (homoskedastic)*

Description

Sargan test of overidentifying restrictions (homoskedastic)

Usage

```
morie_iv_sargan(data, outcome, endogenous, instruments, exogenous = NULL)
```

Arguments

data	A data.frame (or tibble) holding the outcome, endogenous regressors, instruments, and any exogenous controls.
outcome	Character; column name of the response variable.
endogenous	Character vector; column names of the endogenous regressors.
instruments	Character vector; column names of the instrumental variables.
exogenous	Optional character vector of additional exogenous regressors included in both the structural equation and the first stage. NULL (default) for a just-identified design.

---

morie_iv_split_sample *Split-sample IV*

Description

Split-sample IV

Usage

```
morie_iv_split_sample(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  split_fraction = 0.5,
  seed = 42,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`split_fraction`	Fraction of the data used in the first stage.
`seed`	RNG seed.
`alpha`	Significance level for confidence intervals.

`morie_iv_stock_yogo` *Stock-Yogo critical values*

Description

Stock-Yogo critical values

Usage

```
morie_iv_stock_yogo(n_endogenous = 1, n_instruments = 1)
```

Arguments

`n_endogenous`	Integer; number of endogenous regressors used to look up the Stock-Yogo critical-value table.
`n_instruments`	Integer; number of instruments used to look up the Stock-Yogo critical-value table.

`morie_iv_tsls` *Two-Stage Least Squares (2SLS)*

Description

Estimates a linear IV model via 2SLS, preferring `ivreg::ivreg`.

Usage

```
morie_iv_tsls(
  data,
  outcome,
  endogenous,
  instruments,
  exogenous = NULL,
  cluster = NULL,
  robust = TRUE,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Name of the outcome column.
`endogenous`	Character vector of endogenous regressor names.
`instruments`	Character vector of excluded-instrument names.
`exogenous`	Optional character vector of exogenous covariate names.
`cluster`	Optional name of a cluster ID column.
`robust`	Logical; if `TRUE` use HC1 robust standard errors.
`alpha`	Significance level for confidence intervals.

Value

A list with class `morie_iv_result` containing coefficients, standard errors, t-statistics, p-values, confidence interval bounds, variable names, sample size, method label, and a `details` list.

`morie_iv_wald` *Wald (single-instrument) estimator*

Description

$\hat{\beta} = (\bar{y}_{z=1} - \bar{y}_{z=0}) / (\bar{d}_{z=1} - \bar{d}_{z=0})$.

Usage

```
morie_iv_wald(data, outcome, treatment, instrument, alpha = 0.05)
```

Arguments

`data`	Data frame.
`outcome`	Outcome column.
`treatment`	Endogenous treatment column.
`instrument`	Binary instrument column.
`alpha`	Significance level.

`morie_jackknife_estimate` *Delete-1 jackknife variance estimate*

Description

Delete-1 jackknife variance estimate

Usage

```
morie_jackknife_estimate(df, statistic)
```

Arguments

`df`	A data frame.
`statistic`	A function taking a data frame and returning a scalar.

Value

Named list: `estimate`, `se`, `bias`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_johansen_cointegration` *Johansen trace test for cointegration*

Description

Johansen trace test for cointegration

Usage

```
morie_johansen_cointegration(x, k_ar_diff = 1)
```

Arguments

`x`	Numeric matrix (T x k) of I(1) candidate series.
`k_ar_diff`	Number of lagged differences. Default 1.

Value

Named list with `eigenvalues`, `trace_stat`, `crit_values`, `rank`, `n`, `k`, `method`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_kalman_filter` *Kalman filter predict-update for a linear-Gaussian state-space model*

Description

Defaults to a univariate local-level model when matrices are omitted.

Usage

```
morie_kalman_filter(
  x,
  transition = NULL,
  H = NULL,
  Q = NULL,
  R = NULL,
  x0 = NULL,
  P0 = NULL
)
```

Arguments

`x`	Numeric vector or matrix of observations.
`transition`	Transition matrix (default identity).
`H`	Observation matrix (default identity).
`Q`	State-innovation covariance (default sigma^2 I).
`R`	Observation covariance (default sigma^2 I).
`x0`	Initial state mean.
`P0`	Initial state covariance.

Value

Named list with `state`, `state_cov`, `innovations`, `innovation_variance`, `loglik`, `n`, `method`.

Examples

```
morie_kalman_filter(x = rnorm(50))
```

`morie_kendall_tau` *Kendall's tau-b*

Description

Kendall's tau-b

Usage

```
morie_kendall_tau(x, y)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.

Value

Named list: `tau`, `p_value`.

Examples

```
morie_kendall_tau(x = rnorm(50), y = rnorm(50))
```

`morie_kendall_tau_partial` *Kendall partial-tau correlation (Gibbons Ch 12.6)*

Description

tau_xy.z = (tau_xy - tau_xz tau_yz) / sqrt((1 - tau_xz^2)(1 - tau_yz^2))

Usage

```
morie_kendall_tau_partial(x, y, z)
```

Arguments

`x`, `y`, `z`	Numeric vectors of equal length.

Value

Named list: statistic (partial tau), p_value, tau_xy, tau_xz, tau_yz, z, n.

Examples

```
morie_kendall_tau_partial(x = rnorm(50), y = rnorm(50), z = rnorm(50))
```

`morie_kernel_pca` *Kernel principal components analysis via* **kernlab**

Description

Thin extender over `kernlab::kpca` that performs PCA in a feature space induced by a reproducing-kernel.

Usage

```
morie_kernel_pca(x, ...)
```

Arguments

`x`	Numeric matrix or data frame of features (rows = observations).
`...`	Further arguments forwarded to `kernlab::kpca` (e.g. `kernel`, `kpar`, `features`, `th`).

Value

A list with `$method = "kernlab::kpca"` and `$raw` (a `kpca` S4 object with eigenvalues, eigenvectors and the projected rotated data).

Examples

```
## Not run:
  if (requireNamespace("kernlab", quietly = TRUE)) {
    set.seed(1)
    x <- matrix(stats::rnorm(200), ncol = 4)
    morie_kernel_pca(x, kernel = "rbfdot", features = 2)
  }

## End(Not run)
```

---

morie_kmeans_clustering

K-means clustering (R parity)

Description

Wraps `stats::kmeans` with Hartigan-Wong (the default).

Usage

```
morie_kmeans_clustering(
  x,
  n_clusters = 3L,
  n_init = 10L,
  max_iter = 300L,
  seed = 0L
)
```

Arguments

`x`	Numeric matrix.
`n_clusters`	Number of clusters K.
`n_init`	Number of random restarts.
`max_iter`	Max Lloyd iterations.
`seed`	RNG seed.

Value

Named list: estimate (inertia), labels, centers, inertia, n_iter, n_clusters, n, method.

Examples

```
morie_kmeans_clustering(x = rnorm(50))
```

morie_kruskal_wallis_test

Kruskal-Wallis non-parametric ANOVA

Description

Kruskal-Wallis non-parametric ANOVA

Usage

```
morie_kruskal_wallis_test(...)
```

Arguments

...	Numeric vectors, one per group.

Value

Named list: `H`, `df`, `p_value`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_laniyonu_actuarial_risk_disparity

Replication of O'Connell & Laniyonu (2025) — CSC actuarial-risk disparity

Description

R port of `morie.laniyonu.actuarial_risk_disparity`. Audits the Correctional Service of Canada's four ordinal risk instruments (Static, DFIA-R Dynamic, Offender Security Level, Reintegration Potential) and the two binary downstream outcomes (parole granted; institutional housing level) for race x gender bias.

Pass one of the four ordinal risk scores (`"static"`, `"dynamic"`, `"osl"`, `"reintegration"`) to run the Stage 1 threshold-specific ordinal logit; pass `"parole"` or `"housing"` to run the Stage 2 score-net-residual binary logit.

Usage

```
morie_laniyonu_actuarial_risk_disparity(
  df,
  outcome,
  race_cols,
  gender_col = "gender",
  score_col = NULL,
  control_cols = NULL,
  ordinal_levels = c("low", "medium", "high"),
  outcome_col = NULL,
  split_by_gender = TRUE,
  bootstrap_replicates = 200L,
  random_state = 20260513L
)
```

Arguments

`df`	Sentence-level (one row per sentence) CSC microdata.
`outcome`	One of `"static"`, `"dynamic"`, `"osl"`, `"reintegration"`, `"parole"`, `"housing"`.
`race_cols`	Character vector of 0/1 race indicator columns (White is the implicit reference; pass non-reference levels).
`gender_col`	Categorical gender column.
`score_col`	Required when `outcome` is `"parole"` or `"housing"`; the actuarial score column name (e.g. `"reintegration_score_numeric"` for parole).
`control_cols`	Optional additional control columns (age, priors, sentence length, etc.). Pre-dummy any categoricals.
`ordinal_levels`	Level ordering for ordinal outcomes (default `c("low", "medium", "high")`).
`outcome_col`	Optional override for the default column name.
`split_by_gender`	If `TRUE` (default), stratifies by `gender_col`; if `FALSE`, folds one-hot gender into the design.
`bootstrap_replicates`	Stage 2 only; bootstrap reps for residual race-coefficient SEs.
`random_state`	Seed for bootstrap.

Details

Two empirical stages match the paper:

- Stage 1 (ordinal scores): *threshold-specific* cumulative-logit, fit as two separate binary logits at the low->medium and medium->high cutoffs, plus a proportional-odds LR test. The headline pattern is much larger |beta| at the low->medium cut than at medium->high.
- Stage 2 (binary outcomes): the score-net-residual audit - logistic regression of outcome on actuarial score + race indicators (+ controls). A non-zero residual race coefficient is the disparate-treatment signal.

Caveat surfaced via `warning()` on every call (Goel et al. 2021): a non-zero residual race coefficient is evidence of OUTPUT disparity, not PREDICTIVE-VALIDITY disparity. The two are conceptually distinct; the paper's disparate-treatment claim rests on the former.

Value

A named `list` of class `morie_laniyonu_ard_result` carrying the per-stratum coefficients and a multi-paragraph `interpretation` string.

A named `list` of class `morie_laniyonu_ard_result`.

References

O'Connell, C., & Laniyonu, A. (2025). Race, gender, and risk assessments in Canadian federal prison. Race & Justice, 15(3), 428-453.

Goel, S., Shroff, R., Skeem, J., & Slobogin, C. (2021). The accuracy, equity, and jurisprudence of criminal risk assessment. In Research Handbook on Big Data Law (pp. 9-28).

`morie_laniyonu_gentrification_policing`

Replication of Laniyonu (2018) — Coffee Shops and Street Stops

---

Description

R port of `morie.laniyonu.gentrification_policing`. Estimates the direct, indirect (spatial spillover), and total effect of gentrification on NYPD stop-and-frisk rates at the census-tract x year level via a Spatial Durbin Model (SDM) decomposition.

Usage

```
morie_laniyonu_gentrification_policing(
  df,
  year_col = "year",
  tract_id_col = "tract_id",
  stops_col = "stops",
  population_col = "population",
  crime_col = "felony_count",
  demand_col = "calls_311_omp",
  baseline_income_col = "median_inc_2000",
  baseline_rent_col = "median_rent_2000",
  growth_college_col = NULL,
  growth_rent_col = NULL,
  follow_income_col = "median_inc_2014",
  follow_rent_col = "median_rent_2014",
  baseline_college_col = "pct_ba_2000",
  follow_college_col = "pct_ba_2014",
  additional_controls = NULL,
  weight_matrix = NULL,
```

```
  weight_matrix_kind = c("queen", "knn"),
  fitted_rho = NULL,
  fitted_beta_direct = NULL,
  fitted_beta_spatial = NULL,
  years = NULL,
  log_outcome = TRUE
)
```

Arguments

`df`	Tract-year panel. One row per tract per year.
`year_col`, `tract_id_col`, `stops_col`, `population_col`, `crime_col`, `demand_col`	Column names; defaults match the morie toy bundle schema.
`baseline_income_col`, `baseline_rent_col`	Baseline-period income and rent (2000 in the paper).
`growth_college_col`, `growth_rent_col`	Growth columns. If `NULL`, computed from follow-minus-baseline.
`follow_income_col`, `follow_rent_col`, `baseline_college_col`, `follow_college_col`	Used when growth columns are not pre-computed.
`additional_controls`	Extra tract-year controls (pct_black, etc.).
`weight_matrix`	Pre-computed (N, N) row-standardised spatial weights. Required when `fitted_*` mode is in use.
`weight_matrix_kind`	Provenance label only.
`fitted_rho`, `fitted_beta_direct`, `fitted_beta_spatial`	Pre-fitted SDM outputs. Pass these to bypass lite-mode.
`years`	Subset of years to analyse.
`log_outcome`	If `TRUE` (default), outcome is `log(stops / population)`.

Details

The paper's headline finding: gentrification has roughly zero *direct* effect on stops/capita inside the gentrifying tract, but a +51\ stops/capita in neighbouring tracts.

Two modes are supported:

- **Pre-fitted mode (preferred)**: pass `fitted_rho`, `fitted_beta_direct`, `fitted_beta_spatial` from your own SDM fit (e.g.\ **spatialreg::lagsarlm** with Durbin terms). This wrapper handles the diagnostic ladder + Kelejian-Prucha spillover decomposition.
- **Lite mode (fall-back)**: OLS + Moran's I on residuals, decomposition with rho=0. Useful for sanity-checks only.

Value

A `list` of class `morie_laniyonu_gp_result`, one per year analysed.

A `list` of `morie_laniyonu_gp_result`, one per year analysed.

References

Laniyonu, A. (2018). Coffee shops and street stops: Policing practices in gentrifying neighborhoods. Urban Affairs Review, 54(5), 898-930.

LeSage, J. P., & Pace, R. K. (2009). Introduction to Spatial Econometrics. CRC Press.

`morie_laniyonu_smi_force_disparity`
Replication of Laniyonu & Goff (2021) — Police force vs SMI disparity

---

Description

R port of `morie.laniyonu.smi_force_disparity`. Estimates a hierarchical negative-binomial model with a synthetic area-exposure (SAE) offset for persons-with-serious-mental-illness (PwSMI), with year fixed effects and an area random intercept.

Composes a synthetic-area-exposure (SAE) step (base-R logistic on survey microdata, predicted at ACS tract marginals) into a negative-binomial GLM with year fixed effects and an area random intercept approximated by ridge-penalised area dummies.

Usage

```
morie_laniyonu_smi_force_disparity(
  df,
  survey_df,
  survey_trait_col = "smi",
  survey_covariate_cols,
  area_covariate_cols = NULL,
  force_count_col = "force_events",
  non_smi_count_col = NULL,
  geog_col = "tract_id",
  year_col = "year",
  population_col = "pop_18plus",
  baseline_year = NULL,
  include_year_fe = TRUE,
  include_area_re = TRUE,
  max_iter = 500L,
  tol = 1e-06,
  return_design = FALSE
)
```

Arguments

`df`	Force-event panel, one row per (area, year).
`survey_df`	Survey microdata for fitting P(SMI | covariates).
`survey_trait_col`	Binary column in `survey_df`.

`survey_covariate_cols`
: Covariates available in BOTH survey_df and df.

`area_covariate_cols`
: Optional rename map for df.

`force_count_col`
: Count of force events against PwSMI per (area, year).

`non_smi_count_col`
: Count of force events against non-SMI per (area, year). If `NULL`, df must contain `total_force_events` and the non-SMI count is computed as total minus PwSMI.

`geog_col`, `year_col`, `population_col`
: Column names.

`baseline_year`
: Year to drop as the reference (default = min).

`include_year_fe`, `include_area_re`
: Toggle the year FE / area RE blocks.

`max_iter`, `tol`
: Optimiser controls.

`return_design`
: Attach `X`, `y`, `offset` to `exposure_summary` for hand-off to brms / rstanarm.

Details

The trick: there is no administrative census of who has SMI at the tract level, so the denominator is built by:

1. Fitting *P(SMI | age, sex, race, income, ...)* on a national survey using only covariates also tabulated at the tract level by the ACS.
2. Applying those coefficients to ACS tract marginals to get a per-tract predicted P(SMI).
3. Multiplying by adult population for a synthetic exposure denominator n_{vti}.

The count model is

$$y_{vti} \sim \text{NegBin}(n_{vti} \exp(\mu + \alpha_v + \delta_t + \beta_i), \phi)$$

with v = PwSMI vs non-SMI, t = year, i = area. The headline coefficient α_v is the log relative-risk of police use of force against PwSMI vs non-SMI.

Paper headlines: RR PwSMI = 11.6x (tract); 10.2x (precinct).

This R port is a frequentist MLE approximation (via `stats::glm.nb` in MASS, falling back to a hand-rolled NB MLE on `stats::optim` if MASS is unavailable). For paper-grade Bayesian credible intervals, fit in **brms** / **rstanarm** using the design matrix returned with `return_design=TRUE`.

Surfaces a `warning()` on every call: the SMI flag on force events is a proxy biased TOWARD THE NULL (officers miss more SMI than they over-attribute), so the estimated α_v is a **conservative lower bound** on the true disparity.

Value

A `list` of class `morie_laniyonu_smi_result`.

A list of class `morie_laniyonu_smi_result`.

References

Laniyonu, A., & Goff, P. A. (2021). Measuring disparities in police use of force and injury among persons with serious mental illness. BMC Psychiatry, 21(1), 500.

`morie_lcmm_latent_class`

Latent-class mixed models via **lcmm**

Description

Thin extender over `lcmm::lcmm` for the Proust-Lima et al. (2017) latent-class linear mixed model on longitudinal / repeated-measures data.

Usage

```
morie_lcmm_latent_class(fixed, random = ~1, subject, data, ng = 2, ...)
```

Arguments

`fixed`	A two-sided formula for the fixed-effects part of the model.
`random`	A one-sided formula for the random-effects part (default `~1`, random intercept only).
`subject`	Character; the name of the column in `data` identifying the subject / grouping variable.
`data`	A data frame containing the variables in `fixed`, `random`, and `subject`.
`ng`	Integer; the number of latent classes (default `2`).
`...`	Further arguments forwarded to `lcmm::lcmm` (e.g. `mixture`, `classmb`, `idiag`, `nwg`, `link`, `intnodes`, `epsa`, `epsb`, `epsd`, `maxiter`, `B`, `convB`, `convL`, `convG`, `verbose`).

Value

A list with `$method = "lcmm::lcmm"` and `$raw` (an `lcmm` object with the class-membership probabilities, class-specific fixed-effect estimates, and convergence diagnostics).

Examples

```
## Not run:
  if (requireNamespace("lcmm", quietly = TRUE)) {
    data("data_hlme", package = "lcmm")
    morie_lcmm_latent_class(
      fixed   = Y ~ Time,
      random  = ~ Time,
      subject = "ID",
      data    = data_hlme,
      ng      = 2,
      mixture = ~ Time
```

```
  )
}

## End(Not run)
```

`morie_learning_curve` *Learning curve – train / val MSE vs training-set size (R parity)*

Description

Manual implementation of the sklearn morie_learning_curve flow: shuffle, split into k folds, for each train-fraction fit on a prefix of the training fold and score on the held-out fold.

Usage

```
morie_learning_curve(x, y, sizes = NULL, cv = 5L, seed = 0L)
```

Arguments

`x`	Numeric matrix predictors.
`y`	Numeric response.
`sizes`	Training-set fractions (default seq(0.1, 1.0, length=5)).
`cv`	Number of CV folds.
`seed`	RNG seed for shuffling.

Value

Named list: estimate (final val MSE), train_sizes, train_scores, val_scores, n, method.

Examples

```
morie_learning_curve(x = rnorm(50), y = rnorm(50))
```

`morie_levene_test` *Levene test for equality of variances*

Description

Levene test for equality of variances

Usage

```
morie_levene_test(...)
```

Arguments

...	Numeric vectors, one per group.

Value

Named list: `F`, `p_value`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

---

morie_license_metadata

morie's SPDX-style licence metadata

Description

morie's SPDX-style licence metadata

Usage

```
morie_license_metadata()
```

Value

A named list summarising morie's licence posture, useful for pipeline build manifests, auditd logs, and downstream compliance pipelines.

Examples

```
morie_license_metadata()
```

---

morie_linear_regression_ols

Ordinary least squares closed-form solution (R parity)

Description

Wraps `stats::lm` and returns coefficients plus classical OLS standard errors.

Usage

```
morie_linear_regression_ols(x, y)
```

Arguments

x	Numeric matrix or vector of predictors.
y	Numeric response vector.

Value

Named list with `estimate` (intercept + slopes), `se`, `n`, `method`.

References

Hastie, Tibshirani & Friedman, Elements of Statistical Learning (2009).

Examples

```
morie_linear_regression_ols(x = rnorm(50), y = rnorm(50))
```

`morie_list_datasets` *List all datasets with cache status*

Description

List all datasets with cache status

Usage

```
morie_list_datasets(db_path = NULL, con = NULL)
```

Arguments

db_path	Optional path to a SQLite/DuckDB file (default backend).
con	Optional pre-opened DBI connection (overrides `db_path`).

Value

A data.frame with columns: key, name, source, survey, year, type, cached (logical), rows (integer or NA).

Examples

```
morie_list_datasets()
```

`morie_list_morie_modules`
List implemented MORIE CPADS modules

Description

List implemented MORIE CPADS modules

Usage

```
morie_list_morie_modules()
```

Value

Data frame describing the implemented module surface.

Examples

```
morie_list_morie_modules()
```

`morie_llm_agent_available`
Return TRUE when at least one live LLM provider is available

Description

Return TRUE when at least one live LLM provider is available

Usage

```
morie_llm_agent_available()
```

Value

Logical scalar.

`morie_llm_ask` *Send a prompt to the best available LLM provider*

Description

R port of `morie.llm.ask`. Tries each provider in priority order; on HTTP/timeout failure falls through to the next, and finally to a static local help string.

Usage

```
morie_llm_ask(
  prompt,
  context = NULL,
  model = NULL,
  provider = NULL,
  system_prompt = NULL,
  timeout = 120
)
```

Arguments

`prompt`	User question or instruction.
`context`	Optional named list injected as text into the system prompt.
`model`	Optional model override.
`provider`	Optional provider override (ollama/gemini/api/openai/local). NULL = auto-detect.
`system_prompt`	Optional full system-prompt override.
`timeout`	HTTP timeout in seconds. Default 120.

Value

Character scalar response text, or local-fallback text when all providers fail.

`morie_llm_ask_multi` *Ask the best available LLM provider, accepting a multi-turn messages list*

Description

R port of `morie.llm.ask_multi`. Unlike :func:`morie_llm_ask`, this accepts a pre-built `messages` list (each element: `role`/`content`) enabling multi-turn conversation. Streaming is not supported in the R port – this always returns a single character scalar.

Usage

```
morie_llm_ask_multi(messages, providers = NULL, model = NULL, timeout = 120)
```

Arguments

`messages`	list of role/content lists.
`providers`	Optional character vector forcing a specific provider ordering. When NULL the auto-detected provider is tried first, then the remaining providers in priority order.
`model`	Optional model identifier.
`timeout`	HTTP timeout in seconds.

Details

Provider fall-through order mirrors :func:`morie_llm_detect_provider`: ollama -> freeapi -> gemini -> api -> openai -> local.

Value

Character scalar response text.

`morie_llm_detect_provider` *Detect the active LLM provider*

Description

Detect the active LLM provider

Usage

```
morie_llm_detect_provider()
```

Value

Character scalar provider key: ollama / gemini / api / openai / local.

morie_llm_list_freeapi_models

List vendored OllamaFreeAPI model catalogue

Description

R port of `list_freeapi_models`. Walks any JSON files in `inst/ollama_json/` (mirroring the Python `morie/ollama_json/` vendoring) and emits a data.frame with one row per unique model. When the catalogue directory is absent (R-only install), a single fallback row for the default model is returned so downstream callers always get a usable table.

Usage

```
morie_llm_list_freeapi_models()
```

Value

data.frame with columns model / family / size / label / alias.

morie_llm_probe_freeapi

Probe an OllamaFreeAPI community server

Description

R port of the Python `_probe_freeapi` helper. Returns `TRUE` when at least one free remote model is reachable. The result is cached in `options(morie.llm.freeapi_cached)` for the process lifetime. A single one-second retry is performed because community servers can be slow.

Usage

```
morie_llm_probe_freeapi(timeout = 4)
```

Arguments

`timeout`	Probe timeout in seconds. Default 4.

Value

Logical scalar.

`morie_llm_probe_ollama` *Probe a local Ollama instance*

Description

Probe a local Ollama instance

Usage

```
morie_llm_probe_ollama(timeout = 2)
```

Arguments

`timeout`	Probe timeout in seconds.

Value

Logical scalar – TRUE when reachable.

`morie_llm_request_completion` *POST a chat-completion request to an OpenAI-compatible endpoint*

Description

POST a chat-completion request to an OpenAI-compatible endpoint

Usage

```
morie_llm_request_completion(
  base_url,
  model,
  messages,
  api_key = NULL,
  timeout = 120
)
```

Arguments

`base_url`	Provider base URL.
`model`	Model identifier.
`messages`	List of role/content lists.
`api_key`	Optional bearer token (NULL for local Ollama).
`timeout`	Seconds. Default 120.

Value

Parsed JSON list (the response body).

`morie_load_cpads` *Load CPADS data: local files -> cache -> CKAN API*

Description

Resolution order:

1. Local RDS/CSV files in standard project locations
2. SQLite cache (`data/cache/morie.db`)
3. CKAN API fetch (requires internet)

Usage

```
morie_load_cpads(db_path = NULL, use_ckan = TRUE, con = NULL)
```

Arguments

`db_path`	Optional path to a SQLite/DuckDB file (default backend).
`use_ckan`	Logical; if TRUE and data not found locally or in cache, attempt to fetch from the CKAN API.
`con`	Optional pre-opened DBI connection (overrides `db_path`).

Value

A data.frame with canonical CPADS columns.

Examples

```
## Not run:
# Needs the CPADS PUMF (local file, cache, or a live CKAN fetch).
cpads <- morie_load_cpads(use_ckan = TRUE)
if (!is.null(cpads)) head(cpads)

## End(Not run)
```

morie_load_cpads_data *Load the real CPADS CSV from this repository*

Description

Load the real CPADS CSV from this repository

Usage

```
morie_load_cpads_data(cpads_csv = .cpads_default_csv())
```

Arguments

cpads_csv	Path to the CPADS CSV.

Value

Canonicalized CPADS data frame.

Examples

```
# Reads and canonicalises the CPADS PUMF CSV. The default CSV lives in
# a morie project tree; the CKAN-fetched PUMF works identically (see
# morie_load_dataset("ocp21")). The tryCatch guard lets the example
# render cleanly on machines without the CSV checked out locally.
tryCatch(morie_load_cpads_data(), error = function(e) message(conditionMessage(e)))
```

morie_load_dataset *Load a dataset by catalog key*

Description

Resolution tiers, tried in order: built-in DB -> user cache -> local file -> CKAN datastore -> direct download URL -> ArcGIS layer -> error. Supports fuzzy matching: `morie_load_dataset("cpads_2021")` resolves to `ocp21`.

Usage

```
morie_load_dataset(key, db_path = NULL, refresh = FALSE, con = NULL)
```

Arguments

`key`	Dataset catalog key (or fuzzy match).
`db_path`	Optional path to a SQLite/DuckDB file (default backend).
`refresh`	If `TRUE`, bypass the built-in database and the user cache (and, for remotely-backed datasets, the local file) and re-fetch from the remote source, overwriting the cached copy. Use this to pick up time-to-time updates to a dataset.
`con`	Optional pre-opened DBI connection for the user cache (overrides `db_path`). The built-in DB read is always SQLite-based and is unaffected by `con`.

Value

A data.frame.

See Also

`morie_fetch`, `morie_ckan_search`

Examples

```
## Not run:
df <- morie_load_dataset("ocp21") # CPADS 2021-2022 (default DuckDB cache)
df <- morie_load_dataset("ocp21", refresh = TRUE) # force re-fetch

# PostgreSQL cache (run a server first):
# con <- DBI::dbConnect(RPostgres::Postgres(),
#   host = "localhost", dbname = "morie", user = "...")
# df <- morie_load_dataset("ocp21", con = con)

## End(Not run)
```

`morie_locfdr_estimate`

Local FDR estimation via **locfdr**

Description

Thin extender over `locfdr::locfdr` for Efron's empirical-Bayes local false-discovery-rate estimation from a vector of z-scores (Efron, 2004; Efron, 2010).

Usage

```
morie_locfdr_estimate(zz, ...)
```

Arguments

zz	Numeric vector of test statistics (typically z-scores) for the locfdr empirical-null fit.
...	Further arguments forwarded to `locfdr::locfdr` (e.g. `bre`, `df`, `pct`, `pct0`, `nulltype`, `type`, `plot`, `mult`, `mlests`, `main`, `sw`).

Value

A list with `$method = "locfdr::locfdr"` and `$raw` (the `locfdr` object with the fitted local-FDR curve, empirical-null parameters, and the fdr / Fdrleft / Fdrright summaries).

Examples

```
## Not run:
  if (requireNamespace("locfdr", quietly = TRUE)) {
    set.seed(1)
    zz <- c(stats::rnorm(900), stats::rnorm(100, mean = 3))
    morie_locfdr_estimate(zz)
  }

## End(Not run)
```

---

morie_mann_whitney_test

Mann-Whitney U test (Wilcoxon rank-sum)

Description

Mann-Whitney U test (Wilcoxon rank-sum)

Usage

```
morie_mann_whitney_test(
  x1,
  x2,
  alternative = c("two.sided", "greater", "less")
)
```

Arguments

x1	Numeric vector (group 1).
x2	Numeric vector (group 2).
alternative	`"two.sided"`, `"greater"`, or `"less"`.

Value

Named list: `W`, `p_value`, `r` (effect size).

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_marker_variance`

Marker variance-component estimation

Description

Reports sigma_m^2 via the VanRaden split sigma_g^2 / (2 sum p_j q_j) alongside the naive sigma_g^2 / p form. sigma_g^2 is obtained from a quick GBLUP fit.

Usage

```
morie_marker_variance(x, y, markers)
```

Arguments

`x`	Fixed-effect design (optional).
`y`	Numeric response.
`markers`	(n x m) genotype matrix coded 0/1/2.

Value

list(estimate, sigma_g2, sigma_e2, h2, sigma_m2_vanraden, sigma_m2_naive, sum_2pq, p_freq, n, p, method).

References

VanRaden (2008); Montesinos Lopez Ch 3.

Examples

```
morie_marker_variance(
  x = rnorm(50), y = rnorm(50),
  markers = matrix(sample(0:2, 200, TRUE), 50, 4)
)
```

`morie_matching_abadie_imbens_se` *Abadie-Imbens standard error for matching estimators*

Description

Computes the conditional-variance Abadie-Imbens SE accounting for the fact that matching introduces correlation across matched observations.

Usage

```
morie_matching_abadie_imbens_se(
  data,
  outcome,
  treatment,
  match_pairs,
  n_matches = 1L
)
```

Arguments

`data`	Data frame.
`outcome`, `treatment`	Column names.
`match_pairs`	Data frame of matched indices.
`n_matches`	Number of matches per treated unit (carried for parity).

Value

Scalar numeric Abadie-Imbens SE.

References

Abadie, A., & Imbens, G. W. (2006). Large sample properties of matching estimators for average treatment effects. *Econometrica*, 74(1), 235–267.

Examples

```
## Not run:
morie_matching_abadie_imbens_se(df, "y", "d", res$match_pairs)

## End(Not run)
```

`morie_matching_atc_matched` *ATC from a matched sample*

Description

Estimates the Average Treatment Effect on the Controls. Uses the explicit `_matched` suffix to distinguish it from the IPW estimator `morie_estimate_atc` in `causal.R`.

Usage

```
morie_matching_atc_matched(data, outcome, treatment, match_pairs, alpha = 0.05)
```

Arguments

`data`	Data frame.
`outcome`	Outcome column name.
`treatment`	Binary treatment column name.
`match_pairs`	Data frame with columns `treated_idx` and `control_idx`.
`alpha`	Significance level for confidence intervals.

Value

A list of class `morie_te_result`.

Examples

```
## Not run:
morie_matching_atc_matched(df, "y", "d", res$match_pairs)

## End(Not run)
```

`morie_matching_ate_matched` *ATE from a matched / weighted sample*

Description

Estimates the Average Treatment Effect via a (weighted) mean difference between treated and control outcomes. Uses the explicit `_matched` suffix to distinguish it from the IPW estimator `morie_estimate_ate` in `causal.R`.

Usage

```
morie_matching_ate_matched(
  data,
  outcome,
  treatment,
  covariates,
  weights = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`, `treatment`	Column names.
`covariates`	Character vector of covariates (carried for parity with the Python signature).
`weights`	Optional column of matching / weighting weights.
`alpha`	Significance level for confidence intervals.

Value

A list of class `morie_te_result`.

Examples

```
## Not run:
morie_matching_ate_matched(df, "y", "d", c("x1", "x2"),
                           weights = "._cem_weight")

## End(Not run)
```

`morie_matching_att_matched` *ATT from a matched sample*

Description

Estimates the Average Treatment effect on the Treated using paired differences from a matched sample. Uses the explicit `_matched` suffix to distinguish it from the IPW estimator `morie_estimate_att` in `causal.R`.

Usage

```
morie_matching_att_matched(
  data,
  outcome,
  treatment,
  match_pairs,
  weights = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Outcome column name.
`treatment`	Binary treatment column name.
`match_pairs`	Data frame with columns `treated_idx` and `control_idx`.
`weights`	Optional column of matching weights.
`alpha`	Significance level for confidence intervals.

Value

A list of class `morie_te_result`.

Examples

```
## Not run:
res <- morie_matching_nearest_neighbor(df, "d", c("x1", "x2"))
morie_matching_att_matched(df, "y", "d", res$match_pairs)

## End(Not run)
```

`morie_matching_balance` *Balance diagnostics for matched / weighted samples*

Description

Reports standardised mean differences (SMD), variance ratios, and Kolmogorov-Smirnov statistics for each covariate. For a richer covariate-balance report (including continuous + categorical handling and Love-plot rendering), see `cobalt::bal.tab` / `cobalt::love.plot`.

Usage

```
morie_matching_balance(
  data,
  treatment,
  covariates,
  weights = NULL,
  threshold = 0.1
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`weights`	Optional column name of matching / weighting weights.
`threshold`	Absolute-SMD threshold for the `balanced` flag.

Value

A list of class `morie_balance_result` with `balance_table` (a data frame) and scalar summaries `overall_balance`, `max_smd`, `balanced`.

Examples

```
## Not run:
morie_matching_balance(df, "d", c("x1", "x2"))

## End(Not run)
```

`morie_matching_balance_table` *Publication-ready balance table*

Description

Thin wrapper around `morie_matching_balance` returning only the data-frame component. See `cobalt::bal.tab` for an alternative with categorical-variable support.

Usage

```
morie_matching_balance_table(data, treatment, covariates, weights = NULL)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`weights`	Optional column name of matching / weighting weights.

Value

A data frame.

Examples

```
## Not run:
morie_matching_balance_table(df, "d", c("x1", "x2"))

## End(Not run)
```

`morie_matching_cardinality` *Cardinality matching*

Description

Finds the largest matched sample with maximum absolute SMD below `balance_threshold`. Uses an iterative caliper-tightening heuristic over `morie_matching_nearest_neighbor`; for an exact mixed-integer-programming alternative see `designmatch::cardmatch`.

Usage

```
morie_matching_cardinality(
  data,
  treatment,
  covariates,
  balance_threshold = 0.1,
  ps = NULL
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`balance_threshold`	Maximum absolute SMD tolerated (default 0.1).
`ps`	Optional pre-computed propensity scores.

Value

A list of class `morie_match_result`.

References

Zubizarreta, J. R. (2012). Using mixed integer programming for matching in an observational study of kidney failure after surgery. *JASA*, 107(500), 1360–1371.

Examples

```
## Not run:
morie_matching_cardinality(df, "d", c("x1", "x2"),
                           balance_threshold = 0.1)

## End(Not run)
```

`morie_matching_cem` *Coarsened Exact Matching (CEM)*

Description

Thin wrapper around `MatchIt::matchit(method = "cem")`, which in turn calls the **cem** package.

Usage

```
morie_matching_cem(data, treatment, covariates, n_bins = 5L)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`n_bins`	Either a single integer (applied to every covariate) or a named list mapping covariate name to the number of bins (forwarded as `cutpoints`).

Value

A list of class `morie_match_result`.

References

Iacus, S. M., King, G., & Porro, G. (2012). Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20(1), 1–24.

Examples

```
## Not run:
morie_matching_cem(df, "d", c("x1", "x2"), n_bins = 5)

## End(Not run)
```

`morie_matching_common_support` *Restrict a sample to the region of common support*

Description

Drops units whose propensity score falls outside the overlap region of treated and control units.

Usage

```
morie_matching_common_support(
  data,
  treatment,
  ps_col = "propensity_score",
  method = "minmax"
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`ps_col`	Propensity-score column name (default `"propensity_score"`).
`method`	One of `"minmax"` (overlap of ranges) or `"trim"` (drop the extreme 5 percent of each tail).

Value

A subset of `data` on common support.

Examples

```
## Not run:
df$propensity_score <- morie_matching_estimate_propensity(df, "d",
                                                          c("x1", "x2"))
morie_matching_common_support(df, "d")

## End(Not run)
```

`morie_matching_doubly_robust` *Doubly-robust ATT combining matching and regression*

Description

Matches on the propensity score, then applies bias-corrected linear regression adjustment within the matched sample. Standard errors come from a non-parametric bootstrap.

Usage

```
morie_matching_doubly_robust(
  data,
  outcome,
  treatment,
  covariates,
  ps = NULL,
  n_bootstrap = 200L,
  seed = 42L,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`, `treatment`	Column names.
`covariates`	Character vector of covariates.
`ps`	Optional pre-computed propensity scores.
`n_bootstrap`	Number of bootstrap replications.
`seed`	Random seed.
`alpha`	Significance level.

Value

A list of class `morie_te_result` with estimand `"ATT_DR"`.

Examples

```
## Not run:
morie_matching_doubly_robust(df, "y", "d", c("x1", "x2"),
                             n_bootstrap = 200)

## End(Not run)
```

`morie_matching_entropy_balance` *Entropy balancing weights (Hainmueller, 2012)*

Description

Thin wrapper around `WeightIt::weightit(method = "ebal")` (or `ebal::ebalance` if **WeightIt** is unavailable). Computes weights for the control group so that the weighted moments of the covariates match those of the treated group.

Usage

```
morie_matching_entropy_balance(
  data,
  treatment,
  covariates,
  max_moment = 1L,
  max_iter = 500L,
  tol = 1e-06
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`max_moment`	Highest moment to balance (1 = means, 2 = means + var, 3 = + skewness).
`max_iter`	Maximum Newton iterations (forwarded to **ebal**).
`tol`	Convergence tolerance (forwarded to **ebal**).

Value

A numeric vector of weights aligned to the rows of `data` after dropping NAs. Treated units receive weight 1.

References

Hainmueller, J. (2012). Entropy balancing for causal effects. *Political Analysis*, 20(1), 25–46.

Examples

```
## Not run:
w <- morie_matching_entropy_balance(df, "d", c("x1", "x2"))

## End(Not run)
```

`morie_matching_estimate_propensity` *Estimate propensity scores*

Description

Estimates the probability of treatment via logistic regression or gradient boosting on a set of covariates.

Usage

```
morie_matching_estimate_propensity(
  data,
  treatment,
  covariates,
  model = "logistic",
  max_iter = 1000
)
```

Arguments

`data`	Data frame.
`treatment`	Name of the binary treatment column (0/1).
`covariates`	Character vector of covariate names.
`model`	One of `"logistic"` (default) or `"gbm"`. `"gbm"` requires the **gbm** package.
`max_iter`	Maximum iterations for logistic regression.

Value

A numeric vector of propensity scores aligned to the rows of `data` (after dropping NAs in `treatment` or `covariates`); the `names` of the vector are the row names of the retained rows.

References

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1), 41–55.

Examples

```
## Not run:
df <- data.frame(d = rbinom(200, 1, 0.4),
                 x1 = rnorm(200), x2 = rnorm(200))
ps <- morie_matching_estimate_propensity(df, "d", c("x1", "x2"))

## End(Not run)
```

morie_matching_exact *Exact matching on discrete covariates*

Description

Thin wrapper around `MatchIt::matchit(method = "exact")`.

Usage

```
morie_matching_exact(data, treatment, exact_vars)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`exact_vars`	Character vector of discrete variables for exact matching.

Value

A list of class `morie_match_result`.

Examples

```
## Not run:
morie_matching_exact(df, "d", c("region", "year"))

## End(Not run)
```

morie_matching_full *Full matching via subclassification*

Description

Thin wrapper around `MatchIt::matchit(method = "full")`, which calls **optmatch**.

Usage

```
morie_matching_full(data, treatment, covariates, ps = NULL, n_subclasses = 10L)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`ps`	Optional pre-computed propensity scores (ignored; retained for back-compat).
`n_subclasses`	Carried for back-compat (ignored under MatchIt).

Value

A list of class `morie_match_result`.

References

Hansen, B. B. (2004). Full matching in an observational study of coaching for the SAT. *JASA*, 99(467), 609–618.

Examples

```
## Not run:
morie_matching_full(df, "d", c("x1", "x2"))

## End(Not run)
```

`morie_matching_genetic` *Genetic matching (Diamond & Sekhon, 2013)*

Description

Thin wrapper around `Matching::GenMatch` + `Matching::Match` that returns a `morie_match_result`. Uses a genetic algorithm to find covariate weights for Mahalanobis distance matching that maximise covariate balance.

Usage

```
morie_matching_genetic(
  data,
  treatment,
  covariates,
  n_neighbors = 1L,
  pop_size = 50L,
  n_generations = 20L,
  seed = 42L
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`n_neighbors`	Number of matches per treated unit (`M` in **Matching**).
`pop_size`	Genetic-algorithm population size (default 50).
`n_generations`	Number of GA generations.
`seed`	Random seed.

Value

A list of class `morie_match_result`.

References

Diamond, A., & Sekhon, J. S. (2013). Genetic matching for estimating causal effects. *Review of Economics and Statistics*, 95(3), 932–945.

Examples

```
## Not run:
morie_matching_genetic(df, "d", c("x1", "x2"),
                       pop_size = 50, n_generations = 20)

## End(Not run)
```

`morie_matching_longitudinal` *Longitudinal matching for panel data*

Description

Matches treated and control units on the basis of their pre-treatment covariate values.

Usage

```
morie_matching_longitudinal(
  data,
  treatment,
  covariates,
  unit,
  time,
  treatment_time,
  n_pre_periods = 1L,
  method = "nearest_neighbor"
)
```

Arguments

`data`	Panel data frame.
`treatment`	Binary treatment indicator column.
`covariates`	Character vector of covariates.
`unit`	Column name identifying units.
`time`	Column name identifying time.
`treatment_time`	Column giving the (per-unit) start of treatment; non-finite values indicate never-treated.

`n_pre_periods`	Number of pre-treatment periods to summarise.
`method`	One of `"nearest_neighbor"` or `"mahalanobis"`.

Value

A list of class `morie_match_result`.

Examples

```
## Not run:
morie_matching_longitudinal(panel, "d", c("x1"), unit = "id",
                            time = "t", treatment_time = "t0")

## End(Not run)
```

`morie_matching_love_plot_data` *Love-plot data: pre- vs post-matching balance*

Description

Returns a data frame suitable for plotting absolute SMDs before and after matching. For a publication-ready plot, pass the same `matchit` object to `cobalt::love.plot`.

Usage

```
morie_matching_love_plot_data(
  unmatched_data,
  matched_data,
  treatment,
  covariates,
  weights_col = NULL
)
```

Arguments

`unmatched_data`, `matched_data`	Data frames.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`weights_col`	Optional column of matching weights in `matched_data`.

Value

A data frame with columns `covariate`, `smd_before`, `smd_after`, `abs_smd_before`, `abs_smd_after`.

Examples

```
## Not run:
morie_matching_love_plot_data(df, res$matched_data,
                              "d", c("x1", "x2"))

## End(Not run)
```

`morie_matching_mahalanobis`
Mahalanobis distance matching

Description

Thin wrapper around `MatchIt::matchit(distance = "mahalanobis")`.

Usage

```
morie_matching_mahalanobis(
  data,
  treatment,
  covariates,
  n_neighbors = 1L,
  caliper = NULL,
  replace = FALSE,
  exact = NULL
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of continuous covariates.
`n_neighbors`	Number of matches per treated unit.
`caliper`	Maximum Mahalanobis distance for a valid match.
`replace`	If `TRUE`, controls may be re-used.
`exact`	Optional character vector of variables to match exactly prior to distance matching.

Value

A list of class `morie_match_result`.

Examples

```
## Not run:
morie_matching_mahalanobis(df, "d", c("x1", "x2"), n_neighbors = 1)

## End(Not run)
```

`morie_matching_multi_treatment` *Matching with multiple (> 2) treatment groups*

Description

For each non-reference treatment level, matches treated units to the reference group via the chosen binary matching method.

Usage

```
morie_matching_multi_treatment(
  data,
  treatment,
  covariates,
  reference_group = NULL,
  method = "nearest_neighbor"
)
```

Arguments

`data`	Data frame.
`treatment`	Treatment column (may take more than two levels).
`covariates`	Character vector of covariates.
`reference_group`	Optional reference level (defaults to the modal level).
`method`	One of `"nearest_neighbor"` or `"mahalanobis"`.

Value

A named list whose keys are treatment levels and whose values are `morie_match_result` objects.

Examples

```
## Not run:
morie_matching_multi_treatment(df, "treat3", c("x1", "x2"))

## End(Not run)
```

`morie_matching_nearest_neighbor` *Nearest-neighbour propensity-score matching*

Description

Thin wrapper around `MatchIt::matchit(method = "nearest")` that returns a `morie_match_result` list for compatibility with the rmorie balance / treatment-effect helpers.

Usage

```
morie_matching_nearest_neighbor(
  data,
  treatment,
  covariates,
  n_neighbors = 1L,
  caliper = NULL,
  replace = FALSE,
  ps = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column (0/1).
`covariates`	Character vector of covariates for the propensity model.
`n_neighbors`	Number of matches per treated unit (forwarded as `ratio`).
`caliper`	Maximum logit-propensity distance for a valid match, expressed in SD units of the logit (or `NULL` for no caliper).
`replace`	If `TRUE`, controls may be re-used.
`ps`	Optional pre-computed propensity scores (ignored; retained for back-compat).
`alpha`	Significance level (carried through to `details`).

Value

A list of class `morie_match_result`.

Examples

```
## Not run:
res <- morie_matching_nearest_neighbor(df, "d", c("x1", "x2"),
                                       caliper = 0.2)

## End(Not run)
```

`morie_matching_optimal_pair` *Optimal pair matching*

Description

Thin wrapper around `MatchIt::matchit(method = "optimal")`, which calls **optmatch**.

Usage

```
morie_matching_optimal_pair(
  data,
  treatment,
  covariates,
  distance = "propensity",
  ps = NULL
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`distance`	One of `"propensity"` or `"mahalanobis"`.
`ps`	Optional pre-computed propensity scores (ignored; retained for back-compat).

Value

A list of class `morie_match_result`.

Examples

```
## Not run:
morie_matching_optimal_pair(df, "d", c("x1", "x2"))

## End(Not run)
```

morie_matching_overlap

Propensity-score overlap diagnostics

Description

Reports the propensity-score range overlap between treated and control, the number / percentage of units off support, and the IPW effective sample size.

Usage

```
morie_matching_overlap(data, treatment, covariates, ps = NULL)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column.
`covariates`	Character vector of covariates.
`ps`	Optional pre-computed propensity scores.

Value

A list with `ps_summary` (per-group quantiles), `overlap_region`, `n_off_support`, `pct_off_support`, and `effective_sample_size`.

Examples

```
## Not run:
morie_matching_overlap(df, "d", c("x1", "x2"))

## End(Not run)
```

morie_matching_quality

Comprehensive matching-quality assessment

Description

Compares balance before and after matching and reports percent bias reduction, count of balanced covariates, and overlap statistics.

Usage

```
morie_matching_quality(
  unmatched_data,
  matched_data,
  treatment,
  covariates,
  weights = NULL
)
```

Arguments

`unmatched_data`, `matched_data`	Data frames.
`treatment`	Binary treatment column.
`covariates`	Character vector of covariates.
`weights`	Optional column of matching weights in `matched_data`.

Value

A list with `balance_before`, `balance_after`, `bias_reduction`, `mean_bias_reduction`, `pct_balanced_before`, `pct_balanced_after`, `n_obs_before`, `n_obs_after`.

Examples

```
## Not run:
morie_matching_quality(df, res$matched_data, "d", c("x1", "x2"))

## End(Not run)
```

`morie_matching_rosenbaum_bounds`

Rosenbaum bounds for hidden bias

Description

Computes bounds on the p-value for the treatment effect over a grid of values of `gamma` (the maximum odds ratio of differential treatment assignment due to an unobserved confounder). Uses the Wilcoxon signed-rank approach. For exact bounds, see `sensitivitymv::senmv` or the **rbounds** package.

Usage

```
morie_matching_rosenbaum_bounds(
  data,
  outcome,
  treatment,
  match_pairs,
  gamma_range = NULL
)
```

Arguments

`data`	Data frame.
`outcome`, `treatment`	Column names.
`match_pairs`	Data frame of matched indices.
`gamma_range`	Optional numeric vector of Γ values.

Value

A data frame with columns `gamma`, `p_lower`, `p_upper`, `significant_lower`, `significant_upper`.

References

Rosenbaum, P. R. (2002). *Observational Studies* (2nd ed.). Springer.

Examples

```
## Not run:
morie_matching_rosenbaum_bounds(df, "y", "d", res$match_pairs)

## End(Not run)
```

`morie_matching_subclassify` *Subclassification (stratification) on the propensity score*

Description

Thin wrapper around `MatchIt::matchit(method = "subclass")` that reports within-stratum sample sizes and PS ranges, preserving the rmorie return shape (`data_with_strata` + `stratum_effects`).

Usage

```
morie_matching_subclassify(
  data,
  treatment,
  covariates,
  ps = NULL,
  n_strata = 5L
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`ps`	Optional pre-computed propensity scores (ignored; retained for back-compat).
`n_strata`	Number of quantile-based strata (default 5).

Value

A list with components `data_with_strata` (the matched data augmented with `._stratum` and `._ps` columns) and `stratum_effects` (per-stratum sample sizes and PS ranges).

Examples

```
## Not run:
morie_matching_subclassify(df, "d", c("x1", "x2"), n_strata = 5)

## End(Not run)
```

`morie_matching_trim_propensity` *Trim propensity scores to a fixed range*

Description

Clips propensity scores to `[lower, upper]`.

Usage

```
morie_matching_trim_propensity(ps, lower = 0.01, upper = 0.99)
```

Arguments

`ps`	Numeric vector of propensity scores.
`lower`, `upper`	Numeric clip bounds (defaults 0.01, 0.99).

Value

A numeric vector of the same length as `ps`.

Examples

```
morie_matching_trim_propensity(c(0.001, 0.5, 0.999))
```

`morie_matching_variable_ratio` *Variable-ratio matching on propensity score*

Description

Thin wrapper around `MatchIt::matchit(method = "nearest",ratio = max_ratio, min.controls = min_ratio)` which supports variable-ratio nearest-neighbour matching natively.

Usage

```
morie_matching_variable_ratio(
  data,
  treatment,
  covariates,
  min_ratio = 1L,
  max_ratio = 5L,
  caliper = 0.2,
  ps = NULL
)
```

Arguments

`data`	Data frame.
`treatment`	Binary treatment column name.
`covariates`	Character vector of covariates.
`min_ratio`, `max_ratio`	Match-count bounds per treated unit.
`caliper`	Caliper on the propensity score (in SD units).
`ps`	Optional pre-computed propensity scores (ignored; retained for back-compat).

Value

A list of class `morie_match_result`.

Examples

```
## Not run:
morie_matching_variable_ratio(df, "d", c("x1", "x2"),
                              min_ratio = 1, max_ratio = 3)

## End(Not run)
```

`morie_meta_rma` *Random- / fixed-effects meta-analysis via* **metafor**

Description

Thin extender over `metafor::rma` that fits a (possibly moderated) random- or fixed-effects meta-analytic model to per-study effect sizes and their sampling variances.

Usage

```
morie_meta_rma(yi, vi, data = NULL, ...)
```

Arguments

`yi`	Numeric vector of study-level effect-size estimates.
`vi`	Numeric vector of sampling variances corresponding to `yi`.
`data`	Optional data frame to evaluate `yi`, `vi` and any moderators against; passed straight through to `metafor::rma`.
`...`	Further arguments forwarded to `metafor::rma` (e.g. `mods`, `method`, `weights`, `test`, `level`, `slab`).

Value

A list with `$method = "metafor::rma"` and `$raw` (an `rma.uni` object with the pooled estimate, heterogeneity statistics and moderator effects).

Examples

```
## Not run:
  if (requireNamespace("metafor", quietly = TRUE)) {
    set.seed(1)
    k <- 12
    vi <- stats::runif(k, 0.02, 0.10)
    yi <- stats::rnorm(k, mean = 0.3, sd = sqrt(vi))
    morie_meta_rma(yi = yi, vi = vi)
  }

## End(Not run)
```

`morie_midas_regression`
MIDAS regression with Beta-polynomial weights

Description

MIDAS regression with Beta-polynomial weights

Usage

```
morie_midas_regression(x, y, K = NULL)
```

Arguments

`x`	High-frequency regressor matrix (n_t x K) or flat vector.
`y`	Low-frequency target (length n_t).
`K`	Number of high-frequency lags (required when x is flat).

Value

Named list with `beta0`, `beta1`, `theta1`, `theta2`, `weights`, `r2`, `n`, `K`, `method`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_mini_batch_gradient`
Mini-batch stochastic gradient descent for OLS (R parity)

Description

Mini-batch stochastic gradient descent for OLS (R parity)

Usage

```
morie_mini_batch_gradient(
  x,
  y,
  lr = 0.01,
  n_epochs = 200,
  batch_size = 32L,
  seed = 0L
)
```

Arguments

x	Numeric matrix / vector predictors.
y	Numeric response.
lr	Learning rate.
n_epochs	Number of passes over the data.
batch_size	Mini-batch size.
seed	RNG seed for shuffling.

Value

Named list: estimate, reference_ols, n_epochs, batch_size, loss, n, method.

Examples

```
morie_mini_batch_gradient(x = rnorm(50), y = rnorm(50))
```

---

morie_ml_apply_smote *Apply SMOTE oversampling to balance a binary outcome*

Description

R port of `morie.ml.apply_smote`. Uses `smotefamily::SMOTE` when installed and feasible; falls back to random oversampling (duplicate minority rows) otherwise. Returns the resampled `(X, y)` together with a status list of before/after counts and the method used.

Usage

```
morie_ml_apply_smote(X, y, random_state = 42L, k_neighbors = NULL)
```

Arguments

X	Feature data.frame.
y	Binary outcome vector (numeric or factor).
random_state	Integer seed for the random fallback. Default 42.
k_neighbors	Integer or NULL. SMOTE neighbour count; auto-picked to `min(5, minority - 1)` when NULL.

Value

list(X, y, status) where `status` mirrors the Python dict keys (`method`, `minority_before`, `majority_before`, `imbalance_ratio_before`, `total_before`, `total_after`, plus `class_<label>_before` / `class_<label>_after`).

morie_ml_eval_robustness

Evaluate Random Forest robustness

Description

Fits a 100-tree Random Forest on training data and reports a classification report (precision / recall / F1 / support per class) on the held-out test set. Mirrors `morie.ml.eval_robustness`.

Usage

```
morie_ml_eval_robustness(
  X,
  y,
  test_X,
  test_y,
  n_estimators = 100L,
  random_state = 42L
)
```

Arguments

`X`	Training features (data.frame or matrix).
`y`	Training labels (factor or coercible to factor).
`test_X`	Test features.
`test_y`	Test labels.
`n_estimators`	Number of trees. Default 100.
`random_state`	Integer seed. Default 42.

Value

Named list keyed by class label and `accuracy` with precision / recall / f1-score / support per class, mirroring sklearn's `classification_report(output_dict=True)`.

morie_multi_trait_gblup

Multi-trait GBLUP via vec-stacked mixed-model equations

Description

Multi-trait GBLUP via vec-stacked mixed-model equations

Usage

```
morie_multi_trait_gblup(x, y, markers, Sigma_g = NULL, Sigma_e = NULL)
```

Arguments

`x`	Fixed-effect design (vector or matrix).
`y`	Multi-trait response (n x t).
`markers`	Genotype matrix (n x m).
`Sigma_g`	Optional t x t genetic covariance.
`Sigma_e`	Optional t x t residual covariance.

Value

list(estimate, G_hat, B_hat, Sigma_g, Sigma_e, n, t, method).

References

Montesinos Lopez Ch 10.

Examples

```
morie_multi_trait_gblup(
  x = rnorm(50), y = rnorm(50),
  markers = matrix(sample(0:2, 200, TRUE), 50, 4)
)
```

`morie_multiple_testing` *Multiple testing correction and multiplicity-adjusted inference*

Description

R port of `morie.multiple_testing`. Provides p-value adjustment methods controlling the family-wise error rate (FWER) and the false discovery rate (FDR), simultaneous-inference helpers, p-value combination procedures, and gatekeeping / hierarchical testing.

Details

Every adjustment routine returns a `morie_rich_result` list carrying `original`, `adjusted`, `rejected`, `method`, `alpha`, `n_rejected`, `n_tests`, and an `interpretation` paragraph. P-value combination procedures return a list with the test statistic, combined p-value, and interpretation.

FWER and FDR methods delegate to `stats::p.adjust` for the textbook procedures (Bonferroni, Holm, Hochberg, Hommel, Benjamini-Hochberg, Benjamini-Yekutieli). Combined-p tests delegate to **poolr** when installed and fall back to inline math otherwise. `storey_q` / `estimate_pi0` delegate to **qvalue** when installed.

`morie_mvn_with_covariance` *Draw multivariate normal samples under a structured covariance*

Description

Draw multivariate normal samples under a structured covariance

Usage

```
morie_mvn_with_covariance(
  n,
  p,
  rng,
  kernel = c("ar1", "independent", "compound", "toeplitz"),
  rho = 0.5,
  mean = NULL
)
```

Arguments

`n`	Number of samples.
`p`	Dimension.
`rng`	`morie_sync_rng()` environment.
`kernel`	One of `"independent"`, `"ar1"`, `"compound"`, `"toeplitz"`.
`rho`	Correlation parameter.
`mean`	Optional length-p mean vector.

Value

An n x p matrix of samples.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_mvnorm_pmv` *Multivariate normal rectangle probability via* **mvtnorm**

Description

Thin extender over `mvtnorm::pmvnorm` that evaluates the multivariate normal CDF over a hyper-rectangle $[lower, upper]$.

Usage

```
morie_mvnorm_pmv(lower, upper, mean = rep(0, length(lower)), sigma, ...)
```

Arguments

`lower`	Numeric vector of lower integration limits (`-Inf` permitted).
`upper`	Numeric vector of upper integration limits (`Inf` permitted).
`mean`	Numeric mean vector of the same length as `lower` (defaults to a zero vector).
`sigma`	Numeric positive-(semi)definite covariance matrix.
`...`	Further arguments forwarded to `mvtnorm::pmvnorm` (e.g. `corr`, `algorithm`, `keepAttr`).

Value

A list with `$method = "mvtnorm::pmvnorm"` and `$raw` (a numeric scalar with the estimated probability and Monte-Carlo error attributes attached).

Examples

```
## Not run:
  if (requireNamespace("mvtnorm", quietly = TRUE)) {
    set.seed(1)
    S <- matrix(c(1, 0.4, 0.4, 1), 2, 2)
    morie_mvnorm_pmv(
      lower = c(-1, -1), upper = c(1, 1),
      mean = c(0, 0), sigma = S
    )
  }

## End(Not run)
```

morie_mvnorm_sample *Sample from the multivariate normal via* **mvtnorm**

Description

Thin extender over `mvtnorm::rmvnorm` that draws n observations from the multivariate normal distribution with a given mean vector and covariance matrix.

Usage

```
morie_mvnorm_sample(n, mean = rep(0, ncol(sigma)), sigma, ...)
```

Arguments

`n`	Integer; the number of multivariate observations to draw.
`mean`	Numeric vector of length `ncol(sigma)` giving the mean (defaults to a zero vector).
`sigma`	Numeric positive-(semi)definite covariance matrix.
`...`	Further arguments forwarded to `mvtnorm::rmvnorm` (e.g. `method`, `pre0.9_9994`, `checkSymmetry`).

Value

A list with `$method = "mvtnorm::rmvnorm"` and `$raw` (a numeric matrix of dimension $n \times \text{ncol}(\Sigma)$).

Examples

```
## Not run:
  if (requireNamespace("mvtnorm", quietly = TRUE)) {
    set.seed(1)
    S <- matrix(c(1, 0.4, 0.4, 1), 2, 2)
    morie_mvnorm_sample(100, mean = c(0, 0), sigma = S)
  }

## End(Not run)
```

morie_nbeats_basis *N-BEATS-style polynomial + Fourier basis-expansion forecasting*

Description

N-BEATS-style polynomial + Fourier basis-expansion forecasting

Usage

```
morie_nbeats_basis(x, horizon = 1, n_trend = 3, n_season = 5, period = 12)
```

Arguments

`x`	Numeric history.
`horizon`	Forecast horizon. Default 1.
`n_trend`	Polynomial-trend degree. Default 3.
`n_season`	Number of Fourier harmonics. Default 5.
`period`	Seasonal period. Default 12.

Value

Named list with `forecast`, `fitted`, `trend`, `seasonal`, `theta_trend`, `theta_seasonal`, `r2`, `n`, `horizon`, `method`.

Examples

```
morie_nbeats_basis(x = rnorm(50))
```

`morie_np_kernel_reg` *Nonparametric kernel regression via* **np**

Description

Thin extender over `np::npregbw` + `np::npreg` for kernel-smoothed nonparametric regression with data-driven bandwidth selection (Hayfield & Racine, 2008). Runs the bandwidth-selection routine first and then fits the regression using the chosen bandwidths.

Usage

```
morie_np_kernel_reg(formula, data, ...)
```

Arguments

`formula`	A model formula of the form `y ~ x1 + x2 + ...` passed to `np::npregbw`.
`data`	A data frame containing the variables in `formula`.
`...`	Further arguments forwarded to `np::npregbw` (e.g. `bwmethod`, `bwtype`, `ckertype`, `regtype`, `tol`, `ftol`).

Value

A list with `$method = "np::npreg (bws via npregbw)"` and `$raw`, where `$raw` is itself a list with `$bws` (the `rbandwidth` object from `np::npregbw`) and `$fit` (the `npregression` object from `np::npreg`).

Examples

```
## Not run:
  if (requireNamespace("np", quietly = TRUE)) {
    set.seed(1)
    n  <- 50
    df <- data.frame(x = stats::runif(n, -1, 1))
    df$y <- sin(pi * df$x) + stats::rnorm(n, sd = 0.1)
    morie_np_kernel_reg(y ~ x, data = df)
  }

## End(Not run)
```

---

morie_nyc_nypd_socrata_cap_note

Socrata default-API-cap note + pagination wiring

Description

All NYC OpenData SODA2 endpoints apply a default cap of 1,000 rows per request unless an explicit `$limit` (or `$$app_token` for authenticated requests) is supplied. For the NYPD CJ datasets wrapped here that means:

Details

- `morie_datasets_nyc_nypd_arrests_ytd(offline = FALSE)` returns **only 1,000 rows** by default, even though the live feed carries ~69,300 rows.
- Pass `max_features = N` to lift the single-request cap to N rows (Socrata enforces a hard server-side cap of 50,000 rows per request).
- **Pagination (wired in 3OO).** For full pulls over the cap, pass `paginate = TRUE`. morie walks SODA2 `$offset` in `page_size`-row chunks until the server returns a short page (exhausted) or `max_features` is reached. Without an app_token the per-request ceiling is 1,000 rows so `page_size = 1000` is the default; with `page_size = 50000` + `app_token` you can pull the full ~69K-row arrests_ytd feed in two requests. `max_pages` (default 200) is a safety net against runaway pulls.

Worked example:

```
# Full live pull of the YTD arrests feed (~69K rows over ~70 pages).
df <- morie_datasets_nyc_nypd_arrests_ytd(
  offline = FALSE, paginate = TRUE)

# First 5,000 rows only (5 paged requests of 1,000 each).
df <- morie_datasets_nyc_nypd_arrests_ytd(
  offline = FALSE, paginate = TRUE, max_features = 5000L)
```

The bundled fixtures (offline mode) are unaffected – they ship 5 rows each as deterministic sample data, and `max_features` simply truncates the fixture.

morie_odds_ratio_ci *Odds ratio and 95% CI from a 2x2 contingency table*

Description

Odds ratio and 95% CI from a 2x2 contingency table

Usage

```
morie_odds_ratio_ci(table_2x2, alpha = 0.05)
```

Arguments

table_2x2	A 2x2 matrix: rows are treatment, columns are outcome.
alpha	Significance level.

Value

Named list: or, ci_lower, ci_upper, p_value.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_omega_squared *Omega-squared (less biased than eta-squared)*

Description

Omega-squared (less biased than eta-squared)

Usage

```
morie_omega_squared(f_stat, df_between, df_within, n)
```

Arguments

f_stat	F statistic.
df_between	Degrees of freedom (numerator).
df_within	Degrees of freedom (denominator).
n	Total sample size.

Value

Numeric omega-squared.

Examples

```
morie_omega_squared(f_stat = 5.2, df_between = 2, df_within = 87, n = 90)
```

---

morie_one_sample_coverage

One-sample coverage probability (Gibbons Ch 2.11.1)

Description

For an ordered sample the coverages U_i = F(X_(i)) - F(X_(i-1)) are i.i.d. Beta(1, n) under H0. Returns empirical coverages (rank-based) plus the cumulative coverage F(X_(n)) - F(X_(1)).

Usage

```
morie_one_sample_coverage(x)
```

Arguments

`x`	Numeric vector.

Value

Named list: coverages, cumulative, expected, n, sample_min, sample_max, method.

Examples

```
morie_one_sample_coverage(x = rnorm(50))
```

---

morie_one_sample_t_test

One-sample t-test

Description

One-sample t-test

Usage

```
morie_one_sample_t_test(
  x,
  mu0 = 0,
  alternative = c("two.sided", "greater", "less")
)
```

Arguments

`x`	Numeric vector.
`mu0`	Null hypothesis mean (default 0).
`alternative`	`"two.sided"`, `"greater"`, or `"less"`.

Value

Named list: `t`, `df`, `p_value`, `ci`.

Examples

```
morie_one_sample_t_test(x = rnorm(50))
```

---

morie_ordered_alternatives_test

Jonckheere-Terpstra ordered-alternatives test (Gibbons Ch 10.6)

Description

Tests H0: F_1 = ... = F_k against the ordered alternative H1: F_1 <= F_2 <= ... <= F_k. J = sum over i<j of U_ij (Mann-Whitney counts with 1/2 weight for ties).

Usage

```
morie_ordered_alternatives_test(groups)
```

Arguments

`groups`	List of numeric vectors in monotone hypothesised order.

Details

Normal approximation: E_J = (N^2 - sum n_i^2) / 4 Var_J = (N^2 (2N + 3) - sum n_i^2 (2 n_i + 3)) / 72

Value

Named list: statistic, p_value, z, E_J, Var_J, n, k, method.

Examples

```
morie_ordered_alternatives_test(groups = list(rnorm(20), rnorm(20), rnorm(20)))
```

`morie_ordered_categories` *Linear-by-linear association test for ordered categories (Gibbons Ch 14.6.1)*

Description

M^2 = (n - 1) * cor(u, v)^2 ~ chi^2_1 under independence, where (u, v) are row/column scores weighted by cell counts.

Usage

```
morie_ordered_categories(x, row_scores = NULL, col_scores = NULL)
```

Arguments

`x`	r x c contingency table.
`row_scores`	Length-r row scores; default 1..r.
`col_scores`	Length-c col scores; default 1..c.

Value

Named list: statistic (M^2), p_value, df, n, correlation.

Examples

```
morie_ordered_categories(x = rnorm(50))
```

`morie_otis_aipw_ate` *Doubly-robust (AIPW) ATE on OTIS data via cross-fitted nuisances.*

Description

Uses `n_folds` cross-fitting: propensity (logistic ridge) and outcome regression (OLS separately for D=1 and D=0) are fit on K-1 folds and predicted on the held-out fold. The doubly-robust influence function (Robins-Rotnitzky-Zhao 1994) is averaged to yield the ATE.

Usage

```
morie_otis_aipw_ate(
  df,
  treatment,
  outcome,
  covariates,
  n_folds = 5L,
  seed = 123L,
  eps = 0.02
)
```

Arguments

`df`	A data frame containing `treatment`, `outcome`, and all `covariates`.
`treatment`	Name of the binary treatment column.
`outcome`	Name of the (numeric) outcome column.
`covariates`	Character vector of covariate names.
`n_folds`	Number of cross-fitting folds (default 5).
`seed`	Integer seed for the fold partition (default 123).
`eps`	Propensity clip bound (default 0.02).

Value

A `morie_causal_estimate` list.

References

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). *JASA* 89(427), 846-866.

Examples

```
set.seed(1)
n <- 300L
x <- rnorm(n)
d <- rbinom(n, 1, plogis(0.4 * x))
y <- 0.5 * d + x + rnorm(n)
df <- data.frame(d = d, y = y, x = x)
morie_otis_aipw_ate(df, treatment = "d", outcome = "y",
                    covariates = "x", n_folds = 3L)
```

`morie_otis_aipw_superlearner`
SuperLearner-stacked AIPW (not yet ported).

Description

The python version stacks RF + ridge + OLS/logit + mean (and optionally xgboost) via cross-validated convex weights. The R port would require pulling in **SuperLearner** or hand-rolling the stacked-cross-fit construction.

Usage

```
morie_otis_aipw_superlearner(...)
```

Arguments

`...`	Arguments mirroring `morie_otis_aipw_ate()`.

Value

Stops with a `NotYetPorted` message; for the time being, call `morie_otis_aipw_ate()` with the default cross-fit OLS+logit stack.

Examples

```
## Not run:
  morie_otis_aipw_superlearner(df, treatment = "d", outcome = "y",
                                covariates = "x")

## End(Not run)
```

`morie_otis_all_analyses`
Run the full OTIS analysis bundle

Description

Calls `morie_otis_rplace` / `morie_otis_astcmb` / `morie_otis_volat` / `morie_otis_rctrnd` / `morie_otis_otdesc` on `df` for one fiscal `year` and returns a named list of `morie_otis_result` objects. If `out_dir` is supplied, each result is also written to disk as a `.txt` (`format()`) and a `.json` (`jsonlite::toJSON` when available, else `dput`).

Usage

```
morie_otis_all_analyses(df, year, sex = NULL, out_dir = NULL)
```

Arguments

df	OTIS data.frame.
year	Integer fiscal year.
sex	Optional gender filter passed to morie_otis_rplace.
out_dir	Optional output directory. When non-NULL the directory is created if missing.

Details

CRAN-safe: with out_dir = NULL (default) no files are written.

Value

Named list of morie_otis_results.

Examples

```
## Not run:
  df <- morie_otis_load()
  res <- morie_otis_all_analyses(df, year = 2024)

## End(Not run)
```

---

morie_otis_all_analyze

Comprehensive per-dataset analyses for ALL 28 OTIS public-release files

---

Description

R port of morie.otis_all_analyze. Pairs with the OTIS loaders (see ?morie_otis and the b01/c-series/d-series CSV files under data/datasets/OTIS/) and chains the existing MRM-OTIS callables in mrm_otis.R the same way morie_arsau_analyze_* (in mrm_arsau.R) chains the generic MRM-UoF callables.

Details

For every dataset id (b01..d07) this module exposes morie_otis_analyze_<id>(data). Each analyzer returns a named list with class c("morie_otis_analysis_result","morie_rich_result", "list") containing title / summary_lines / tables / interpretation / warnings / payload, mirroring the Python RichResult shape used in src/morie/otis_all_analyze.py.

Cross-year invariant: UniqueIndividual_ID is reassigned every fiscal year (see variable_taxonomy.R). Every analyzer that touches that column is within-year only.

`morie_otis_analyze_a01` *OTIS a01 high-level causal analysis (MatchIt + IRM-DML).*

Description

Wraps the full causal pipeline for the canonical Restrictive Confinement Detailed Dataset: 8-state alert-combo encoding -> MatchIt 1:1 NN PSM -> IRM-DML with RF nuisances -> multi-way clustered SE.

Usage

```
morie_otis_analyze_a01(data = NULL, out_dir = NULL)
```

Arguments

`data`	a01 data.frame (loaded from `a01_restrictive_confinement_detailed_dataset.csv`). Pass `NULL` to indicate "use registered loader" – the R port requires you supply the data because we don't ship the loader side-effect from R.
`out_dir`	Optional output directory.

Value

A `morie_otis_analysis_result`. If the morie causal helpers aren't loaded, returns a "not yet ported" stub.

Examples

```
## Not run:
morie_otis_analyze_a01(otis_a01)

## End(Not run)
```

`morie_otis_analyze_a01_ruhela_alt_age` *a01 alt-T Ruhela: Age 50+ -> vm count.*

Description

a01 alt-T Ruhela: Age 50+ -> vm count.

Usage

```
morie_otis_analyze_a01_ruhela_alt_age(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm_alt_age(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional a01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_a01_ruhela_alt_age()
```

`morie_otis_analyze_a01_ruhela_alt_gender`

a01 alt-T Ruhela: Female -> vm count.

Description

a01 alt-T Ruhela: Female -> vm count.

Usage

```
morie_otis_analyze_a01_ruhela_alt_gender(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm_alt_gender(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional a01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_a01_ruhela_alt_gender()
```

`morie_otis_analyze_a01_ruhela_alt_toronto`
a01 alt-T Ruhela: Toronto region -> vm count.

Description

a01 alt-T Ruhela: Toronto region -> vm count.

Usage

```
morie_otis_analyze_a01_ruhela_alt_toronto(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm_alt_toronto(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional a01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_a01_ruhela_alt_toronto()
```

`morie_otis_analyze_a01_ruhela_formulations`
OTIS a01 Ruhela formulations (full DLRM).

Description

Runs the complete OTIS-RC methodology arc (IPW + AIPW + g-comp + PSM-NN + PSM-subclass + IRM-DML + match_first + ATC + PLR + SuperLearner) on the canonical alert-complexity -> regional- volatility formulation.

Usage

```
morie_otis_analyze_a01_ruhela_formulations(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_dlrm(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm(data = NULL, out_dir = NULL)
```

Arguments

data	Optional a01 data.frame.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:
morie_otis_analyze_a01_ruhela_formulations(otis_a01)

## End(Not run)
```

---

morie_otis_analyze_a01_ruhela_per_year

Per-year full-DLRM on a01 canonical formulation.

Description

Per-year full-DLRM on a01 canonical formulation.

Usage

```
morie_otis_analyze_a01_ruhela_per_year(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm_per_year(data = NULL, out_dir = NULL)
```

Arguments

data	Optional a01 data.frame.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_a01_ruhela_per_year()
```

`morie_otis_analyze_a01_ruhela_subgroup_female`
a01 subgroup Ruhela: Female-only cell frame.

Description

a01 subgroup Ruhela: Female-only cell frame.

Usage

```
morie_otis_analyze_a01_ruhela_subgroup_female(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm_subgroup_female(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional a01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_a01_ruhela_subgroup_female()
```

`morie_otis_analyze_a01_ruhela_subgroup_male`
a01 subgroup Ruhela: Male-only cell frame.

Description

a01 subgroup Ruhela: Male-only cell frame.

Usage

```
morie_otis_analyze_a01_ruhela_subgroup_male(data = NULL, out_dir = NULL)

morie_otis_analyze_a01_mrm_subgroup_male(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional a01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_a01_ruhela_subgroup_male()
```

---

morie_otis_analyze_a01_with_csi_context

OTIS a01 causal pipeline + Toronto Crime Severity Index context.

Description

Wires together `morie_otis_analyze_a01` (causal IRM-DML) with the Toronto Police Service / StatsCan CSI context. The R port requires the morie causal pipeline and TPS-CSI helpers to be loaded; otherwise returns a "not yet ported" stub.

Usage

```
morie_otis_analyze_a01_with_csi_context(
  data = NULL,
  variant = "total",
  rebase_to_year = 2023L,
  out_dir = NULL
)
```

Arguments

`data`	Optional a01 data.frame.
`variant`	CSI variant: `"total"` or `"violent"`.
`rebase_to_year`	Anchor year for the CSI index column (default 2023). Use `NULL` to skip rebasing.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:
morie_otis_analyze_a01_with_csi_context(otis_a01)

## End(Not run)
```

morie_otis_analyze_all

Run every OTIS analyzer against a named list of datasets

Description

Counterpart to Python `analyze_all()`.

Usage

```
morie_otis_analyze_all(datasets, out_dir = NULL)
```

Arguments

`datasets`	Named list `list(b01 = <df>, c01 = <df>, ...)`. IDs absent from the list are skipped silently.
`out_dir`	Optional directory to write per-dataset `overlay_<id>.rds` files. `NULL` means in-memory only.

Value

Named list of `morie_otis_analysis_result`s.

morie_otis_analyze_b01

Person-level segregation-placement analysis (b01)

Description

Person-level segregation-placement analysis (b01)

Usage

```
morie_otis_analyze_b01(data)
```

Arguments

`data`	b01 data.frame (76,934 rows in the public release).

Value

A `morie_otis_analysis_result` list with reason / alert / year-trend tables. Within-year only – `UniqueIndividual_ID` is not cross-year-safe.

`morie_otis_analyze_b01_ruhela_alt_age`
b01 alt-T Ruhela: Age 50+ -> vm count.

Description

b01 alt-T Ruhela: Age 50+ -> vm count.

Usage

```
morie_otis_analyze_b01_ruhela_alt_age(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm_alt_age(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional b01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b01_ruhela_alt_age()
```

`morie_otis_analyze_b01_ruhela_alt_gender`
b01 alt-T Ruhela: Female -> vm count.

Description

b01 alt-T Ruhela: Female -> vm count.

Usage

```
morie_otis_analyze_b01_ruhela_alt_gender(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm_alt_gender(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional b01 data.frame.
`out_dir`	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b01_ruhela_alt_gender()
```

---

morie_otis_analyze_b01_ruhela_alt_toronto

b01 alt-T Ruhela: Toronto region -> vm count.

Description

b01 alt-T Ruhela: Toronto region -> vm count.

Usage

```
morie_otis_analyze_b01_ruhela_alt_toronto(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm_alt_toronto(data = NULL, out_dir = NULL)
```

Arguments

data	Optional b01 data.frame.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b01_ruhela_alt_toronto()
```

---

morie_otis_analyze_b01_ruhela_formulations

OTIS b01 Ruhela formulations (full DLRM).

Description

OTIS b01 Ruhela formulations (full DLRM).

Usage

```
morie_otis_analyze_b01_ruhela_formulations(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_dlrm(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm(data = NULL, out_dir = NULL)
```

Arguments

data	Optional b01 data.frame.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:
morie_otis_analyze_b01_ruhela_formulations(otis_b01)

## End(Not run)
```

---

morie_otis_analyze_b01_ruhela_per_year

Per-year full-DLRM on b01 canonical formulation.

Description

Per-year full-DLRM on b01 canonical formulation.

Usage

```
morie_otis_analyze_b01_ruhela_per_year(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm_per_year(data = NULL, out_dir = NULL)
```

Arguments

data	Optional b01 data.frame.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b01_ruhela_per_year()
```

`morie_otis_analyze_b01_ruhela_subgroup_female`
b01 subgroup Ruhela: Female-only cell frame.

Description

b01 subgroup Ruhela: Female-only cell frame.

Usage

```
morie_otis_analyze_b01_ruhela_subgroup_female(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm_subgroup_female(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional b01 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b01_ruhela_subgroup_female()
```

`morie_otis_analyze_b01_ruhela_subgroup_male`
b01 subgroup Ruhela: Male-only cell frame.

Description

b01 subgroup Ruhela: Male-only cell frame.

Usage

```
morie_otis_analyze_b01_ruhela_subgroup_male(data = NULL, out_dir = NULL)

morie_otis_analyze_b01_mrm_subgroup_male(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional b01 data.frame.
`out_dir`	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b01_ruhela_subgroup_male()
```

---

morie_otis_analyze_b02

Aggregate segregation days per person per year (b02)

Description

Aggregate segregation days per person per year (b02)

Usage

```
morie_otis_analyze_b02(data)
```

Arguments

data	b02 data.frame.

---

morie_otis_analyze_b02_ruhela_alt_age

b02 alt-T Ruhela: Age 50+ -> total seg days.

Description

b02 alt-T Ruhela: Age 50+ -> total seg days.

Usage

```
morie_otis_analyze_b02_ruhela_alt_age(data = NULL, out_dir = NULL)

morie_otis_analyze_b02_mrm_alt_age(data = NULL, out_dir = NULL)
```

Arguments

data	Optional b02 data.frame.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b02_ruhela_alt_age()
```

`morie_otis_analyze_b02_ruhela_alt_region`
b02 alt-T Ruhela: Toronto region -> total seg days.

Description

b02 alt-T Ruhela: Toronto region -> total seg days.

Usage

```
morie_otis_analyze_b02_ruhela_alt_region(data = NULL, out_dir = NULL)

morie_otis_analyze_b02_mrm_alt_region(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional b02 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b02_ruhela_alt_region()
```

`morie_otis_analyze_b02_ruhela_formulations`
OTIS b02 Ruhela formulations: T=Female -> seg-day count.

Description

OTIS b02 Ruhela formulations: T=Female -> seg-day count.

Usage

```
morie_otis_analyze_b02_ruhela_formulations(data = NULL, out_dir = NULL)

morie_otis_analyze_b02_dlrm(data = NULL, out_dir = NULL)

morie_otis_analyze_b02_mrm(data = NULL, out_dir = NULL)
```

Arguments

`data`	Optional b02 data.frame.
`out_dir`	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:
morie_otis_analyze_b02_ruhela_formulations(otis_b02)

## End(Not run)
```

---

morie_otis_analyze_b03

Segregation placements by alert x institution (b03)

Description

Segregation placements by alert x institution (b03)

Usage

```
morie_otis_analyze_b03(data)
```

Arguments

data	b03 data.frame.

---

morie_otis_analyze_b03_ruhela_aggregate

b03 aggregate Ruhela: Alert presence -> seg placements.

Description

b03 aggregate Ruhela: Alert presence -> seg placements.

Usage

```
morie_otis_analyze_b03_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_b03_mrm_aggregate(data, out_dir = NULL)
```

Arguments

data	b03 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b03_ruhela_aggregate(otis_b03)
```

---

morie_otis_analyze_b04

Placement durations by region & gender (b04)

Description

Placement durations by region & gender (b04)

Usage

```
morie_otis_analyze_b04(data)
```

Arguments

data	b04 data.frame.

---

morie_otis_analyze_b04_ruhela_aggregate

b04 aggregate Ruhela: Female -> median seg duration.

Description

b04 aggregate Ruhela: Female -> median seg duration.

Usage

```
morie_otis_analyze_b04_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_b04_mrm_aggregate(data, out_dir = NULL)
```

Arguments

data	b04 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b04_ruhela_aggregate(otis_b04)
```

`morie_otis_analyze_b05`
Distribution of placements by binned duration (b05)

Description

Distribution of placements by binned duration (b05)

Usage

```
morie_otis_analyze_b05(data)
```

Arguments

`data`	b05 data.frame.

`morie_otis_analyze_b05_mandela_classification`
Mandela-RF on b05 – per-placement Mandela classification by year.

Description

Applies the Sprott-Doob 15-day Mandela threshold to OTIS b05 (Ontario provincial segregation placement counts by binned duration).

Usage

```
morie_otis_analyze_b05_mandela_classification(data, out_dir = NULL)
```

Arguments

`data`	b05 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b05_mandela_classification(otis_b05)
```

morie_otis_analyze_b05_ruhela_aggregate

b05 aggregate Ruhela: schema-no-demographic guard.

Description

OTIS b05 (segregation placements by consecutive duration) does not carry a demographic treatment variable – the published schema is just `EndFiscalYear, Consecutive_Duration,Number_SegregationP` The "Ruhela formulation" presumes a binary treatment column (typically Gender, Race, or alert status) for the aggregate RF test, so b05 has no meaningful aggregate Ruhela analysis on its own. Returns a structured "not applicable" wrapper rather than erroring so dispatcher loops over the b03..b09 family stay green.

Usage

```
morie_otis_analyze_b05_ruhela_aggregate(data, out_dir = NULL)
```

Arguments

`data`	b05 data.frame.
`out_dir`	Optional output directory (unused, accepted for parity with sibling aggregators).

Value

`morie_otis_analysis_result` carrying a "not applicable" note in `warnings`.

Examples

```
## Not run:  morie_otis_analyze_b05_ruhela_aggregate(otis_b05)
```

morie_otis_analyze_b06

Reasons for placement x institution x gender (b06)

Description

Reasons for placement x institution x gender (b06)

Usage

```
morie_otis_analyze_b06(data)
```

Arguments

`data`	b06 data.frame.

`morie_otis_analyze_b06_ruhela_aggregate`
b06 aggregate Ruhela: Disciplinary reason -> seg placements.

Description

b06 aggregate Ruhela: Disciplinary reason -> seg placements.

Usage

```
morie_otis_analyze_b06_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_b06_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	b06 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b06_ruhela_aggregate(otis_b06)
```

`morie_otis_analyze_b07`
Alerts x gender (b07)

Description

Alerts x gender (b07)

Usage

```
morie_otis_analyze_b07(data)
```

Arguments

`data`	b07 data.frame.

`morie_otis_analyze_b07_ruhela_aggregate`
b07 aggregate Ruhela (pivot to long): With-alert -> seg placements.

Description

b07 aggregate Ruhela (pivot to long): With-alert -> seg placements.

Usage

```
morie_otis_analyze_b07_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_b07_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	b07 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b07_ruhela_aggregate(otis_b07)
```

`morie_otis_analyze_b08`
Durations by institution & gender (b08)

Description

Durations by institution & gender (b08)

Usage

```
morie_otis_analyze_b08(data)
```

Arguments

`data`	b08 data.frame.

`morie_otis_analyze_b08_ruhela_aggregate` *b08 aggregate Ruhela: Female -> median seg duration (institution-clustered).*

Description

b08 aggregate Ruhela: Female -> median seg duration (institution-clustered).

Usage

```
morie_otis_analyze_b08_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_b08_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	b08 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_b08_ruhela_aggregate(otis_b08)
```

`morie_otis_analyze_b09` *Individuals by number of placements x gender (b09)*

Description

Individuals by number of placements x gender (b09)

Usage

```
morie_otis_analyze_b09(data)
```

Arguments

`data`	b09 data.frame.

morie_otis_analyze_b09_ruhela_aggregate

b09 aggregate Ruhela: Female -> individuals in segregation.

Description

b09 aggregate Ruhela: Female -> individuals in segregation.

Usage

```
morie_otis_analyze_b09_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_b09_mrm_aggregate(data, out_dir = NULL)
```

Arguments

data	b09 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_b09_ruhela_aggregate(otis_b09)
```

morie_otis_analyze_c_chi2

MRM chi-square family on c-series.

Description

Pearson chi-square + Cramer's V on every meaningful 2-way slice of the c-series datasets. Honour to Prof. Doob's chi-square tradition in Canadian corrections research.

Usage

```
morie_otis_analyze_c_chi2(
  datasets,
  contingency_value = "NumberIndividuals_RestrictiveConfinement",
  out_dir = NULL
)
```

Arguments

`datasets`	Named list of c-series data.frames (e.g. `list(c03 = otis_c03, c04 = otis_c04, ...)`).
`contingency_value`	Count column to pivot on (default `"NumberIndividuals_RestrictiveConfinement"`).
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:
morie_otis_analyze_c_chi2(list(c03 = otis_c03, c04 = otis_c04))

## End(Not run)
```

`morie_otis_analyze_c01` *Total individuals x custody/RC/seg x gender (c01)*

Description

Total individuals x custody/RC/seg x gender (c01)

Usage

```
morie_otis_analyze_c01(data)
```

Arguments

`data`	c01 data.frame.

`morie_otis_analyze_c01_ruhela_aggregate` *c01 aggregate Ruhela: Female -> RC count.*

Description

c01 aggregate Ruhela: Female -> RC count.

Usage

```
morie_otis_analyze_c01_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c01_mrm_aggregate(data, out_dir = NULL)
```

Arguments

data	c01 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_c01_ruhela_aggregate(otis_c01)
```

---

morie_otis_analyze_c01_ruhela_aggregate_region_cluster

c01 region-cluster variant (year-clustered GEE).

Description

c01 region-cluster variant (year-clustered GEE).

Usage

```
morie_otis_analyze_c01_ruhela_aggregate_region_cluster(data, out_dir = NULL)

morie_otis_analyze_c01_mrm_aggregate_region_cluster(data, out_dir = NULL)
```

Arguments

data	c01 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:
morie_otis_analyze_c01_ruhela_aggregate_region_cluster(otis_c01)

## End(Not run)
```

`morie_otis_analyze_c02`
Individuals in RC/seg by institution (c02)

Description

Individuals in RC/seg by institution (c02)

Usage

```
morie_otis_analyze_c02(data)
```

Arguments

`data`	c02 data.frame.

`morie_otis_analyze_c02_ruhela_aggregate`
c02 aggregate Ruhela: Female -> RC (institution GEE).

Description

c02 aggregate Ruhela: Female -> RC (institution GEE).

Usage

```
morie_otis_analyze_c02_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c02_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c02 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c02_ruhela_aggregate(otis_c02)
```

`morie_otis_analyze_c03`
Individuals x race x gender (c03)

Description

Individuals x race x gender (c03)

Usage

```
morie_otis_analyze_c03(data)
```

Arguments

`data`	c03 data.frame.

`morie_otis_analyze_c03_ruhela_aggregate`
c03 aggregate Ruhela: Indigenous -> RC.

Description

c03 aggregate Ruhela: Indigenous -> RC.

Usage

```
morie_otis_analyze_c03_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c03_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c03 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c03_ruhela_aggregate(otis_c03)
```

`morie_otis_analyze_c04`
Individuals in RC/seg by race x region (c04)

Description

Individuals in RC/seg by race x region (c04)

Usage

```
morie_otis_analyze_c04(data)
```

Arguments

`data`	c04 data.frame from OTIS.

Value

RichResult with summary + race-by-region crosstab.

`morie_otis_analyze_c04_ruhela_aggregate`
c04 aggregate Ruhela: Indigenous -> RC (by region).

Description

c04 aggregate Ruhela: Indigenous -> RC (by region).

Usage

```
morie_otis_analyze_c04_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c04_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c04 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c04_ruhela_aggregate(otis_c04)
```

`morie_otis_analyze_c04_ruhela_aggregate_region_cluster`
c04 region-cluster variant.

Description

c04 region-cluster variant.

Usage

```
morie_otis_analyze_c04_ruhela_aggregate_region_cluster(data, out_dir = NULL)

morie_otis_analyze_c04_mrm_aggregate_region_cluster(data, out_dir = NULL)
```

Arguments

`data`	c04 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:
morie_otis_analyze_c04_ruhela_aggregate_region_cluster(otis_c04)

## End(Not run)
```

`morie_otis_analyze_c05`
Individuals in RC/seg by religion x region (c05)

Description

Individuals in RC/seg by religion x region (c05)

Usage

```
morie_otis_analyze_c05(data)
```

Arguments

`data`	c05 data.frame from OTIS.

Value

RichResult with summary + religion-by-region crosstab.

`morie_otis_analyze_c05_ruhela_aggregate`
c05 aggregate Ruhela: non-majority religion -> RC.

Description

c05 aggregate Ruhela: non-majority religion -> RC.

Usage

```
morie_otis_analyze_c05_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c05_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c05 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c05_ruhela_aggregate(otis_c05)
```

`morie_otis_analyze_c06`
Individuals in RC/seg by age category x region (c06)

Description

Individuals in RC/seg by age category x region (c06)

Usage

```
morie_otis_analyze_c06(data)
```

Arguments

`data`	c06 data.frame from OTIS.

Value

RichResult with summary + age-by-region crosstab.

`morie_otis_analyze_c06_ruhela_aggregate`
c06 aggregate Ruhela: Age 50+ -> RC.

Description

c06 aggregate Ruhela: Age 50+ -> RC.

Usage

```
morie_otis_analyze_c06_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c06_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c06 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c06_ruhela_aggregate(otis_c06)
```

`morie_otis_analyze_c07`
Individuals x alerts x gender (c07)

Description

Individuals x alerts x gender (c07)

Usage

```
morie_otis_analyze_c07(data)
```

Arguments

`data`	c07 data.frame.

morie_otis_analyze_c07_ruhela_aggregate

c07 aggregate Ruhela: Alert presence x Gender -> RC.

Description

c07 aggregate Ruhela: Alert presence x Gender -> RC.

Usage

```
morie_otis_analyze_c07_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c07_mrm_aggregate(data, out_dir = NULL)
```

Arguments

data	c07 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_c07_ruhela_aggregate(otis_c07)
```

morie_otis_analyze_c08

Individuals by religion x gender (c08)

Description

Individuals by religion x gender (c08)

Usage

```
morie_otis_analyze_c08(data)
```

Arguments

data	c08 data.frame from OTIS.

Value

RichResult with summary + religion-by-gender crosstab.

`morie_otis_analyze_c08_ruhela_aggregate`
c08 aggregate Ruhela: non-majority religion x gender -> RC.

Description

c08 aggregate Ruhela: non-majority religion x gender -> RC.

Usage

```
morie_otis_analyze_c08_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c08_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c08 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c08_ruhela_aggregate(otis_c08)
```

`morie_otis_analyze_c09`
Individuals by age category x gender (c09)

Description

Individuals by age category x gender (c09)

Usage

```
morie_otis_analyze_c09(data)
```

Arguments

`data`	c09 data.frame from OTIS.

Value

RichResult with summary + age-by-gender crosstab.

`morie_otis_analyze_c09_ruhela_aggregate`
c09 aggregate Ruhela: Age 50+ x gender -> RC.

Description

c09 aggregate Ruhela: Age 50+ x gender -> RC.

Usage

```
morie_otis_analyze_c09_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c09_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c09 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c09_ruhela_aggregate(otis_c09)
```

`morie_otis_analyze_c10`
RC/seg aggregate durations by institution (c10)

Description

RC/seg aggregate durations by institution (c10)

Usage

```
morie_otis_analyze_c10(data)
```

Arguments

`data`	c10 data.frame.

`morie_otis_analyze_c10_ruhela_aggregate`
c10 aggregate Ruhela: Female -> median RC days (institution GEE).

Description

c10 aggregate Ruhela: Female -> median RC days (institution GEE).

Usage

```
morie_otis_analyze_c10_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c10_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c10 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c10_ruhela_aggregate(otis_c10)
```

`morie_otis_analyze_c11`
Individuals by aggregate-duration bin (c11)

Description

Individuals by aggregate-duration bin (c11)

Usage

```
morie_otis_analyze_c11(data)
```

Arguments

`data`	c11 data.frame.

`morie_otis_analyze_c11_mandela_classification`
Mandela-RF on c11 – per-individual Mandela classification by year.

Description

Applies the 15-day threshold to OTIS c11 (Ontario provincial counts of INDIVIDUALS by binned aggregate duration). Reports both restrictive-confinement and segregation-only views.

Usage

```
morie_otis_analyze_c11_mandela_classification(data, out_dir = NULL)
```

Arguments

`data`	c11 data.frame.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_c11_mandela_classification(otis_c11)
```

`morie_otis_analyze_c11_ruhela_aggregate`
c11 aggregate Ruhela: long-duration bin (>=16 days) -> RC.

Description

c11 aggregate Ruhela: long-duration bin (>=16 days) -> RC.

Usage

```
morie_otis_analyze_c11_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c11_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	c11 data.frame.
`out_dir`	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_c11_ruhela_aggregate(otis_c11)
```

---

morie_otis_analyze_c12

RC/seg aggregate durations by region & gender (c12)

Description

RC/seg aggregate durations by region & gender (c12)

Usage

```
morie_otis_analyze_c12(data)
```

Arguments

data	c12 data.frame.

---

morie_otis_analyze_c12_ruhela_aggregate

c12 aggregate Ruhela: Female -> median RC days (by region).

Description

c12 aggregate Ruhela: Female -> median RC days (by region).

Usage

```
morie_otis_analyze_c12_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_c12_mrm_aggregate(data, out_dir = NULL)
```

Arguments

data	c12 data.frame.
out_dir	Optional.

Value

morie_otis_analysis_result.

Examples

```
## Not run:  morie_otis_analyze_c12_ruhela_aggregate(otis_c12)
```

morie_otis_analyze_d_chi2

MRM chi-square family on d-series.

Description

Yearly trend (d01 Poisson CIs) + Alert x Cause / Housing contingency chi^2 + Cramer's V on d06 / d07.

Usage

```
morie_otis_analyze_d_chi2(datasets, out_dir = NULL)
```

Arguments

`datasets`	Named list with `d01`, `d06`, `d07` data.frames.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:
morie_otis_analyze_d_chi2(list(d01 = otis_d01,
                               d06 = otis_d06, d07 = otis_d07))

## End(Not run)
```

morie_otis_analyze_d01

Person-level custodial deaths (d01)

Description

Person-level custodial deaths (d01)

Usage

```
morie_otis_analyze_d01(data)
```

Arguments

`data`	d01 data.frame.

`morie_otis_analyze_d02`
Custodial deaths by gender (d02)

Description

Custodial deaths by gender (d02)

Usage

```
morie_otis_analyze_d02(data)
```

Arguments

`data`	d02 data.frame from OTIS.

Value

RichResult with summary + deaths-by-gender crosstab.

`morie_otis_analyze_d02_ruhela_aggregate`
d02 aggregate Ruhela: Female -> custodial deaths.

Description

d02 aggregate Ruhela: Female -> custodial deaths.

Usage

```
morie_otis_analyze_d02_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_d02_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	d02 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_d02_ruhela_aggregate(otis_d02)
```

`morie_otis_analyze_d03`
Custodial deaths by race (d03)

Description

Custodial deaths by race (d03)

Usage

```
morie_otis_analyze_d03(data)
```

Arguments

`data`	d03 data.frame from OTIS.

Value

RichResult with summary + deaths-by-race crosstab.

`morie_otis_analyze_d03_ruhela_aggregate`
d03 aggregate Ruhela: Indigenous -> custodial deaths.

Description

d03 aggregate Ruhela: Indigenous -> custodial deaths.

Usage

```
morie_otis_analyze_d03_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_d03_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	d03 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_d03_ruhela_aggregate(otis_d03)
```

`morie_otis_analyze_d04`
Custodial deaths by religion (d04)

Description

Custodial deaths by religion (d04)

Usage

```
morie_otis_analyze_d04(data)
```

Arguments

`data`	d04 data.frame from OTIS.

Value

RichResult with summary + deaths-by-religion crosstab.

`morie_otis_analyze_d04_ruhela_aggregate`
d04 aggregate Ruhela: non-majority religion -> custodial deaths.

Description

d04 aggregate Ruhela: non-majority religion -> custodial deaths.

Usage

```
morie_otis_analyze_d04_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_d04_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	d04 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_d04_ruhela_aggregate(otis_d04)
```

`morie_otis_analyze_d05`
Custodial deaths by age category (d05)

Description

Custodial deaths by age category (d05)

Usage

```
morie_otis_analyze_d05(data)
```

Arguments

`data`	d05 data.frame from OTIS.

Value

RichResult with summary + deaths-by-age crosstab.

`morie_otis_analyze_d05_ruhela_aggregate`
d05 aggregate Ruhela: Age 50+ -> custodial deaths.

Description

d05 aggregate Ruhela: Age 50+ -> custodial deaths.

Usage

```
morie_otis_analyze_d05_ruhela_aggregate(data, out_dir = NULL)

morie_otis_analyze_d05_mrm_aggregate(data, out_dir = NULL)
```

Arguments

`data`	d05 data.frame.
`out_dir`	Optional.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_d05_ruhela_aggregate(otis_d05)
```

`morie_otis_analyze_d06`
Custodial deaths by alert x medical cause (d06)

Description

Custodial deaths by alert x medical cause (d06)

Usage

```
morie_otis_analyze_d06(data)
```

Arguments

`data`	d06 data.frame from OTIS.

Value

RichResult with summary + medical-cause-by-alert crosstab.

`morie_otis_analyze_d07`
Custodial deaths by alert x housing unit (d07)

Description

Custodial deaths by alert x housing unit (d07)

Usage

```
morie_otis_analyze_d07(data)
```

Arguments

`data`	d07 data.frame from OTIS.

Value

RichResult with summary + housing-unit-by-alert crosstab.

`morie_otis_analyze_otis_mandela_provincial_vs_federal`
Mandela-RF cross-comparison: Ontario provincial vs federal SIU.

Description

Cross-references the c11 Mandela classification against the Sprott-Doob Feb 2021 federal SIU figures (Table 19, N=1960).

Usage

```
morie_otis_analyze_otis_mandela_provincial_vs_federal(data, out_dir = NULL)
```

Arguments

`data`	c11 data.frame (used to derive the provincial figures).
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:  morie_otis_analyze_otis_mandela_provincial_vs_federal(otis_c11)
```

`morie_otis_analyze_ruhela_grid`
Aggregate Ruhela grid: one-page IRR comparison across analyzers.

Description

Runs every aggregate Ruhela formulation analyzer against the supplied named datasets list and presents a single primary-IRR comparison table (GEE cluster-robust > NB GLM > Poisson GLM).

Usage

```
morie_otis_analyze_ruhela_grid(datasets, out_dir = NULL)
```

Arguments

`datasets`	Named list keyed by dataset id (b03..d05).
`out_dir`	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:
morie_otis_analyze_ruhela_grid(list(b03 = otis_b03, c01 = otis_c01))

## End(Not run)
```

---

morie_otis_analyze_ruhela_master

Paper-ready master report – every Ruhela formulation in one result.

Description

Sections:

1. Aggregate Ruhela formulations grid
2. (Optional) per-row Ruhela formulations on a01/b01/b02
3. MRM chi-square family on c-series + d-series
4. Mandela-RF cross-comparison (provincial vs federal)

Usage

```
morie_otis_analyze_ruhela_master(
  datasets,
  include_per_row = FALSE,
  out_dir = NULL
)
```

Arguments

datasets	Named list of OTIS data.frames.
include_per_row	Logical; if TRUE also runs the slow per-row RFs. Default FALSE.
out_dir	Optional output directory.

Value

morie_otis_analysis_result.

Examples

```
## Not run:
morie_otis_analyze_ruhela_master(datasets_list)

## End(Not run)
```

`morie_otis_analyze_ruhela_per_year` *Per-fiscal-year full-DLRM Ruhela formulation driver.*

Description

Runs the complete 10-estimator DLRM separately on each fiscal year. This is a heavy operation (~7x the single-year runtime).

Usage

```
morie_otis_analyze_ruhela_per_year(
  data,
  ds_id,
  treatment,
  outcome,
  covariates,
  year_col = "EndFiscalYear",
  cluster_col = "EndFiscalYear",
  out_dir = NULL
)
```

Arguments

`data`	Long-format data.frame with treatment / outcome / cov.
`ds_id`	Dataset id label.
`treatment`	Treatment column name.
`outcome`	Outcome column name.
`covariates`	Character vector of covariate column names.
`year_col`	Year column (default `"EndFiscalYear"`).
`cluster_col`	Cluster axis for SE, or `NULL`.
`out_dir`	Optional output directory.

Value

`morie_otis_analysis_result`.

Examples

```
## Not run:
morie_otis_analyze_ruhela_per_year(df, ds_id = "a01",
  treatment = "T", outcome = "Y", covariates = c("Gender"))

## End(Not run)
```

morie_otis_analyzers *Registry of dataset-id -> analyzer*

Description

Mirrors `_ANALYSES` in `src/morie/otis_all_analyze.py`.

Usage

```
morie_otis_analyzers()
```

morie_otis_astcmb *Alert-state combination encoding (8 combos -> complexity index)*

Description

Encodes 3 binary alert flags into 8 combinations (a1..a8) and aggregates per (id, fiscal year), computing a complexity index `ac` = number of distinct combinations observed.

Usage

```
morie_otis_astcmb(
  df,
  alert_cols = c("mental_health_alert", "suicide_risk_alert", "suicide_watch_alert"),
  id_col = "unique_individual_id",
  year_col = "end_fiscal_year"
)
```

Arguments

`df`	data.frame with the three alert columns and id/year cols.
`alert_cols`	Character-3 of alert column names. Default mirrors the lower-case Python schema.
`id_col`, `year_col`	Column names.

Value

`morie_otis_result`.

`morie_otis_causal_grid` *Run IPW / AIPW / IRM-DML on the three canonical (T, Y) pairs.*

Description

Returns one row per (pair, estimator) combination with the ATE, SE, 95% CI, and per-row notes. The IRM-DML row uses the ATE component of `morie_otis_irm_dml()`'s output (not the ATTE / ATC). Concordance across all three estimators is the strongest evidence of an identified causal effect under conditional exchangeability.

Usage

```
morie_otis_causal_grid(df = NULL, seed = 123L)
```

Arguments

`df`	OTIS placement-level data.frame. If `NULL`, attempts to resolve via `morie_otis_load()`.
`seed`	Integer seed for the cross-fitting (default 123).

Value

Data.frame with columns `pair`, `estimator`, `n`, `p_treat`, `ate`, `ate_se`, `ate_pval`, `ci95_lo`, `ci95_hi`, `notes`.

Examples

```
## Not run:
  morie_otis_causal_grid()

## End(Not run)
```

`morie_otis_churn` *Goffmanian institutional-churn analyses on OTIS*

Description

Eleven callables operationalising Goffman's "total institution" framework (Goffman 1961) on the OTIS dataset:

Details

- `morie_otis_repeat_placement_concentration(b09)`
- `morie_otis_within_year_placement_count(b01)`
- `morie_otis_within_year_region_diversity(b01)`
- `morie_otis_mortification_cooccurrence(b01)`
- `morie_otis_disciplinary_medical_overlap(b01)`
- `morie_otis_embedding_distribution(b02)`
- `morie_otis_intra_year_transition_matrix(a01)`
- `morie_otis_path_complexity_gini(b01)`
- `morie_otis_region_alert_state_richness(b01)`
- `morie_otis_regC_demog_contingency(b01)`
- `morie_otis_irr_glmm_vm(b01)`: Poisson + NB2 IRR (requires **MASS** for the negative-binomial fit; falls back to Poisson-only when **MASS** is unavailable).

All metrics are intra-fiscal-year by construction. OTIS `UniqueIndividual_ID` is anonymised as `YYYY-XXXXX-AA`, randomly reassigned each fiscal year and each dataset file, so longitudinal individual-level and cross-dataset linkage are impossible by design (see `docs/methods/otis_linkage.md`). The `variable_taxonomy.R` registry enforces this with `cross_year_safe = FALSE`.

References

Goffman, E. (1961). *Asylums: Essays on the social situation of mental patients and other inmates.* Anchor Books.

Hill, B. M. (1975). A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, 3(5), 1163-1174.

Clauset, A., Shalizi, C. R., & Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4), 661-703.

`morie_otis_churn_analyze_all`

Run all 11 OTIS-churn analyses

Description

Calls every `morie_otis_*` churn callable on its respective input `data.frame` and returns a named list of results. Each input is independent: pass `NULL` (or omit) to skip a metric. If `out_dir` is supplied, each result is also serialised to disk.

Usage

```
morie_otis_churn_analyze_all(
  b01 = NULL,
  b02 = NULL,
  b09 = NULL,
  a01 = NULL,
  out_dir = NULL
)
```

Arguments

b01, b02, b09, a01	Input data.frames (any may be NULL).
out_dir	Optional output directory.

Details

CRAN-safe: with out_dir = NULL no files are written.

Value

Named list of morie_otis_result.

---

morie_otis_classify_mandela_combo

Mandela alert-state classifier for an OTIS placement row.

Description

Encodes the bitfield of three binary alert flags (MentalHealth, SuicideRisk, SuicideWatch) into the 8-state combo label used by Ruhela's primary RF and maps the alert profile to a Mandela-rule category (*compliant* / *at-risk* / *torture*). This wraps the duration-aware morie_siu_classify_mandela() when a row's NumberConsecutiveDays_Segregation is supplied; with the default flags-only mode the categorisation is "alert-only" and degrades gracefully when duration is missing.

Usage

```
morie_otis_classify_mandela_combo(
  mh,
  sr,
  sw,
  days = NA_real_,
  hours_per_day = 22
)
```

Arguments

mh	Integer/logical 0/1 mental-health flag.
sr	Integer/logical 0/1 suicide-risk flag.
sw	Integer/logical 0/1 suicide-watch flag.
days	Optional numeric consecutive-days-segregation. If supplied, delegates to morie_siu_classify_mandela() for the prolonged-solitary determination.
hours_per_day	Optional numeric daily hours in segregation (default 22, the OTIS Restrictive Confinement convention).

Details

Mandela Rule 43-45 thresholds: > 15 consecutive days segregation = prolonged solitary = torture (UN GA 70/175). Anything >= 22h/day for >= 15 days is the canonical torture-eligible band.

Value

Named list with `combo` (integer 0..7), `combo_label` (one of `a1..a8`), `alert_count` (0..3), `mandela_category`, and `notes`.

Examples

```
morie_otis_classify_mandela_combo(1, 0, 0)
morie_otis_classify_mandela_combo(1, 1, 1, days = 20, hours_per_day = 23)
```

---

morie_otis_disciplinary_medical_overlap

Disciplinary x medical-protection overlap

Description

Goffman's "tinkering trades" tension: same person classified by both punitive and therapeutic rationales. Detects any `SegReason_Disciplinary*` flag co-occurring with any `SegReason_*Medical*` flag.

Usage

```
morie_otis_disciplinary_medical_overlap(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

`morie_otis_embedding_distribution`
Total-days embedding distribution (lognormal vs Pareto vs exp)

Description

Fits lognormal, Pareto, and exponential distributions to `TotalAggregatedDays_Segregation` by AIC and reports which family wins.

Usage

```
morie_otis_embedding_distribution(df)
```

Arguments

`df`	b02 data.frame.

Value

`morie_otis_result`.

`morie_otis_intra_year_transition_matrix`
Intra-year region-to-region transition matrix

Description

Markov transition matrix on `Region_AtTimeOfPlacement` within each person-year, with stationary distribution and off-diagonal Theil-T concentration.

Usage

```
morie_otis_intra_year_transition_matrix(df)
```

Arguments

`df`	a01 data.frame.

Value

`morie_otis_result`.

`morie_otis_ipw_ate` *Hajek-stabilised IPW estimator of the ATE on OTIS data.*

Description

Fits a logistic-regression propensity model on `covariates`, clips propensities to $[\varepsilon, 1-\varepsilon]$, and computes the Hajek-normalised difference of weighted means. SE follows the Lunceford-Davidian (2004) sandwich influence-function form.

Usage

```
morie_otis_ipw_ate(df, treatment, outcome, covariates, eps = 0.02)
```

Arguments

`df`	A data frame containing `treatment`, `outcome`, and all `covariates` (rows with NAs in these columns are dropped).
`treatment`	Name of the binary treatment column. Strings `"Yes"/"No"` (case-insensitive) are auto-binarised; numeric columns are coerced with NA -> 0.
`outcome`	Name of the (numeric) outcome column.
`covariates`	Character vector of covariate names. Character / factor columns are converted to drop-first dummies.
`eps`	Numeric in $(0, 0.5)$; propensity clip bound (default 0.02).

Value

A `morie_causal_estimate` list with `estimator`, `ate`, `ate_se`, `ate_pval`, `ate_ci95`, `n`, `n_treated`, `p_treat`, `notes`.

References

Lunceford, J. K. & Davidian, M. (2004). *Statistics in Medicine* 23(19), 2937-2960.

Examples

```
set.seed(1)
n <- 300L
x <- rnorm(n)
d <- rbinom(n, 1, plogis(0.4 * x))
y <- 0.5 * d + x + rnorm(n)
df <- data.frame(d = d, y = y, x = x)
morie_otis_ipw_ate(df, treatment = "d", outcome = "y",
                   covariates = "x")
```

`morie_otis_irm_dml` *Interactive Regression Model DML on OTIS data (ATE, ATTE, ATC).*

Description

Computes the doubly-robust ATE / ATTE / ATC via the Chernozhukov et al. (2018) IRM score with cross-fitted nuisance models. Delegates to **DoubleML**'s `DoubleMLIRM` when the package (with **mlr3** + **mlr3learners**) is installed; otherwise falls back to a self-contained cross-fit using `.otis_logit_fit` for the propensity and OLS for the per-arm outcome regressions (mirroring the python module's `ml_outcome="ols", ml_propensity="logit"` branch).

Usage

```
morie_otis_irm_dml(
  df,
  treatment,
  outcome,
  covariates,
  cluster_cols = NULL,
  n_folds = 3L,
  seed = 123L,
  eps = 0.02,
  match_first = FALSE,
  match_caliper_sd = 0.2
)
```

Arguments

`df`	A data frame.
`treatment`	Binary treatment column name.
`outcome`	Outcome column name.
`covariates`	Character vector of covariate names.
`cluster_cols`	`NULL`, a single column name, or a length-2 character vector for two-way clustering.
`n_folds`	Number of cross-fitting folds (default 3).
`seed`	Integer seed (default 123).
`eps`	Propensity clip bound (default 0.02).
`match_first`	Logical; if `TRUE`, pre-match the sample with 1:1 NN PSM before fitting (default FALSE).
`match_caliper_sd`	Caliper width (default 0.2 * SD of logit-e).

Details

Cluster-robust SE: pass `cluster_cols` as the name (one-way) or character vector (multi-way Cameron-Gelbach-Miller 2011, up to 2-way). `cluster_cols = NULL` gives the heteroskedasticity-consistent SE.

Optional `match_first = TRUE` runs 1:1 nearest-neighbour propensity-score matching on `logit(e(X))` with caliper `match_caliper_sd * SD(logit(e))` first, then fits IRM-DML on the matched subset. Mirrors the MatchIt-then-DML pipeline of OTIS-RC/notez1a.qmd.

Value

Named list with `ate`, `ate_se`, `ate_pval`, `ate_ci95`, `atte`, `atte_se`, `atte_pval`, `atte_ci95`, `atc`, `atc_se`, `atc_pval`, `atc_ci95`, `n`, `n_treated`, `p_treat`, `se_kind`.

References

Chernozhukov, V. et al. (2018). *Econometrics Journal* 21(1), C1-C68. Cameron, A. C., Gelbach, J. B., & Miller, D. L. (2011). *JBES* 29(2), 238-249.

Examples

```
set.seed(1)
n <- 300L
x <- rnorm(n)
d <- rbinom(n, 1, plogis(0.4 * x))
y <- 0.5 * d + x + rnorm(n)
df <- data.frame(d = d, y = y, x = x, id = sample.int(50, n,
                                                      replace = TRUE))
morie_otis_irm_dml(df, treatment = "d", outcome = "y",
                   covariates = "x", n_folds = 3L)
```

---

morie_otis_irr_glmm_vm

Poisson + Negative-Binomial IRR for volatility ~ alert complexity

---

Description

Builds the (id x fiscal year) cell with outcome `vm` (number of distinct regions visited) and treatment `T_high_ac` = 1 if the person-year alert-complexity `ac >= 2`, then fits Poisson and (optionally) negative-binomial GLMs adjusting for Year, Gender, and Age. The NB fit uses **MASS::glm.nb** when available; if not, only Poisson is reported.

Usage

```
morie_otis_irr_glmm_vm(df)
```

Arguments

`df`	b01 data.frame.

Details

No random effect / cluster-robust SE – for paper-grade inference, use the dedicated OTIS DML pipeline.

Value

`morie_otis_result`.

`morie_otis_load` *Load the canonical OTIS CSV*

Description

Reads the Rscript-exported mirror at `file.path(morie_cache_dir("otis"), "otis_main.csv")` unless `csv_path` is supplied. The expected schema is 10 columns: `end_fiscal_year`, `unique_individual_id`, `region_at_time_of_placement`, `region_most_recent_placement`, `gender`, `age_category`, `mental_health_alert`, `suicide_risk_alert`, `suicide_watch_alert`, `number_of_placements`.

Usage

```
morie_otis_load(csv_path = NULL, use_readr = FALSE)
```

Arguments

`csv_path`	Optional explicit CSV path.
`use_readr`	If `TRUE` and **readr** is installed, use `readr::read_csv`; otherwise base `read.csv`. Default `FALSE` (base R for CRAN portability).

Details

To refresh the cache from the canonical `correctional_stats_report_environment.RData` fixture, run the repository script `scripts/export_otis_csv.R`.

Value

data.frame.

See Also

`morie_cache_dir`.

Examples

```
## Not run:
  df <- morie_otis_load()

## End(Not run)
```

morie_otis_make_pair_a

Pair (a): MentalHealth_Alert -> SuicideRisk_Alert (binary -> binary).

Description

The clinical-alert chain: do mental-health flags causally elevate subsequent suicide-risk-alert occurrence, conditional on demographics and region?

Usage

```
morie_otis_make_pair_a(df)
```

Arguments

`df`	OTIS placement-level data.frame.

Value

Named list `list(data, T = "T_a", Y = "Y_a", covariates)`.

Examples

```
## Not run:
  morie_otis_make_pair_a(morie_otis_load())

## End(Not run)
```

morie_otis_make_pair_alert_to_volatility_a01

a01-aware wrapper for the Ruhela alert -> volatility builder.

Description

Same as `morie_otis_make_pair_alert_to_volatility_ruhela()` but auto-loads a01 (Restrictive Confinement Detailed) via the registered `morie_otis_load()` loader when `df = NULL`. a01 is the canonical file the published OTIS-RC `res_pool` / `res_by_year` / `res_all` are computed on.

Usage

```
morie_otis_make_pair_alert_to_volatility_a01(df = NULL)
```

Arguments

`df`	Optional OTIS a01 data.frame. If `NULL`, attempts to resolve via `morie_otis_load()`.

Value

Same shape as `morie_otis_make_pair_alert_to_volatility_ruhela()`.

Examples

```
## Not run:
  morie_otis_make_pair_alert_to_volatility_a01()

## End(Not run)
```

`morie_otis_make_pair_alert_to_volatility_all` *Run both Ruhela and Naive alert -> volatility builders.*

Description

Convenience wrapper that returns both formulations side-by-side for RDF (Ruhela Dual Formulation) robustness analyses.

Usage

```
morie_otis_make_pair_alert_to_volatility_all(df)
```

Arguments

`df`	OTIS placement-level data.frame.

Value

Named list `list(ruhela = ..., naive = ...)` where each element is the output of the corresponding make-pair builder.

Examples

```
## Not run:
  morie_otis_make_pair_alert_to_volatility_all(morie_otis_load())

## End(Not run)
```

`morie_otis_make_pair_alert_to_volatility_naive` *Naive (max-simultaneous-flags + binary vm) alert -> volatility builder.*

Description

Robustness alternative to `morie_otis_make_pair_alert_to_volatility_ruhela()`: treatment = "max simultaneous flags across the year's rows >= 2"; outcome = "any placement row with regA != regB" (binary). Produces a different treatment marginal (~24\ encoding) and a binary rather than count outcome. Used side-by-side as the Naive arm of an RDF (Ruhela Dual Formulation).

Usage

```
morie_otis_make_pair_alert_to_volatility_naive(df)
```

Arguments

`df`	OTIS placement-level data.frame.

Value

Named list with `data`, `T` = "T_high_ac", `Y` = "Y_vm_any", `covariates` = c("Gender", "Age_Category", "EndFiscalYear").

Examples

```
## Not run:
  df <- morie_otis_load()
  morie_otis_make_pair_alert_to_volatility_naive(df)

## End(Not run)
```

`morie_otis_make_pair_alert_to_volatility_ruhela` *Ruhela's primary alert-complexity -> regional-volatility builder.*

Description

Implements Ruhela's "ac >= 2 -> vm" RF (Ruhela Formulation): the 8-state combo encoding documented in OTIS-RC/notez1a.qmd and used for the published `res_pool` / `res_by_year` / `res_all` estimates. Per `(UniqueIndividual_ID, EndFiscalYear)`, the alert-state *complexity* `ac` is the number of distinct alert combos with positive support across that person-year's rows (NOT the max of simultaneous flags – see the Naive arm for that alternative). Treatment `T_high_ac = 1L` iff `ac >= 2`. Outcome `Y_vm_count` sums the within-row and across-row regional-volatility-move indicators.

Usage

```
morie_otis_make_pair_alert_to_volatility_ruhela(df)
```

Arguments

`df`	OTIS placement-level data.frame (b01 / a01 schema).

Value

A named list with elements `data` (the person-year data.frame), `T` = "T_high_ac", `Y` = "Y_vm_count", and `covariates` = c("Gender", "Age_Category", "EndFiscalYear").

Examples

```
## Not run:
  df <- morie_otis_load()
  pair <- morie_otis_make_pair_alert_to_volatility_ruhela(df)
  morie_otis_irm_dml(pair$data, treatment = pair$T,
                     outcome = pair$Y, covariates = pair$covariates)

## End(Not run)
```

`morie_otis_make_pair_b`

Pair (b): HighAlertComplexity -> AnyReadmission.

Description

Treatment T_b = 1 iff at least 2 of (MentalHealth, SuicideRisk, SuicideWatch) alerts are simultaneously active in the row. Outcome Y_b = 1 iff Number_Of_Placements >= 2 (proxy for any future readmission).

Usage

```
morie_otis_make_pair_b(df)
```

Arguments

`df`	OTIS placement-level data.frame.

Value

Named list `list(data, T = "T_b", Y = "Y_b", covariates)`.

Examples

```
## Not run:
  morie_otis_make_pair_b(morie_otis_load())

## End(Not run)
```

`morie_otis_make_pair_c`
Pair (c): RegionalVolatility -> SegregationDays.

Description

Treatment T_c = 1 iff Region_AtTimeOfPlacement != Region_MostRecent. Outcome Y_c = NumberConsecutiveDays_Segregation winsorised at the 99th percentile.

Usage

```
morie_otis_make_pair_c(df)
```

Arguments

`df`	OTIS placement-level data.frame.

Value

Named list `list(data, T = "T_c", Y = "Y_c", covariates)`.

Examples

```
## Not run:
  morie_otis_make_pair_c(morie_otis_load())

## End(Not run)
```

`morie_otis_mortification_cooccurrence`
Mortification co-occurrence (concurrent alerts)

Description

Counts concurrent alert flags per placement and tests independence of MentalHealth vs SuicideRisk via chi-square + Cramer's V.

Usage

```
morie_otis_mortification_cooccurrence(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

morie_otis_otdesc *Full OTIS descriptive statistics suite*

Description

Returns unique-individual counts overall and by fiscal year, region, age category, and gender, plus the per-individual placement-count five-number summary.

Usage

```
morie_otis_otdesc(
  df,
  id_col = "unique_individual_id",
  year_col = "end_fiscal_year"
)
```

Arguments

`df`	data.frame.
`id_col`, `year_col`	Column names.

Value

`morie_otis_result`.

morie_otis_otdml *Cross-fitted partially linear DML (ATE/ATT) on OTIS*

Description

Wraps a Frisch-Waugh-Lovell partialling-out estimator with `n_folds` cross-fitting on the OLS nuisance functions $E[Y|X]$ and $E[D|X]$, then regresses outcome residuals on treatment residuals for the ATE; heteroskedasticity-robust standard errors. ATT is the ATE divided by the treated share (a simple weighting approximation; for the production-grade DML use **DoubleML**).

Usage

```
morie_otis_otdml(
  df,
  outcome = "Y",
  treatment = "D",
  covariates = NULL,
  n_folds = 3L,
  seed = 123L
)
```

Arguments

`df`	data.frame.
`outcome`, `treatment`	Column names.
`covariates`	Character vector of covariate column names. If `NULL`, defaults to the standard OTIS set.
`n_folds`	Integer fold count (default `3L`).
`seed`	Integer RNG seed.

Details

Categorical covariates are dummy-coded with `model.matrix`.

Value

`morie_otis_result`.

References

Chernozhukov, V. et al. (2018). Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal*, 21(1), C1-C68.

`morie_otis_path_complexity_gini` *Path complexity Gini by (year, region)*

Description

Per-(id, year, region) placement counts, with the Gini coefficient reported overall and split by fiscal year and region.

Usage

```
morie_otis_path_complexity_gini(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

morie_otis_pipeline *OTIS analysis pipeline (RichResult-style driver)*

Description

Wraps the six OTIS primitives in `otis.R` as morie RichResult lists and exposes:

Details

- `morie_otis_load`: canonical CSV loader (reads the Rscript-exported `otis_main.csv` mirror).
- `morie_otis_all_analyses`: driver that runs rplace / astcmb / volat / rctrnd / otdesc on one data.frame and optionally serialises each result to disk under a user-supplied directory (CRAN-safe: never writes without an explicit `out_dir`).

`morie_otis_otdml` is excluded from the bundle because it requires the caller to specify `(treatment, outcome,covariates)` – call it directly when needed.

Year-lock invariant:

OTIS `UniqueIndividual_ID` is randomly reassigned every fiscal year. All analyses are computed within `EndFiscalYear`; cross-year ID joins are forbidden (the `variable_taxonomy.R` registry sets `cross_year_safe = FALSE`).

morie_otis_plr *Partially Linear Regression DML on OTIS (not yet ported).*

Description

The python version uses scikit-learn RF nuisance models for the Frisch-Waugh-Lovell partialling-out construction. For the R port, use the analogous `morie_estimate_double_ml()` from `causal.R`, which already wraps **DoubleML::DoubleMLPLR** (with mlr3 ranger learners) and a cross-fit ridge fallback.

Usage

```
morie_otis_plr(...)
```

Arguments

`...`	Arguments mirroring `morie_otis_aipw_ate()`.

Value

Stops with a redirect to `morie_estimate_double_ml()`.

Examples

```
## Not run:
  morie_otis_plr(df, treatment = "d", outcome = "y",
                 covariates = "x")

## End(Not run)
```

`morie_otis_primitives` *Ontario Restrictive Confinement (OTIS) primitive analyses*

Description

Six lightweight callables mirroring the Python module `morie.otis`: regional-placement matrices, alert-state combo encoding, regional volatility, restrictive-confinement trends, descriptive statistics, and a partialled-out Plug-in DML (PLR) ATE/ATT estimator. Each public callable returns a named list with classes `c("morie_otis_result", "morie_rich_result", "list")` carrying `summary_lines`, optional `tables`, a plain-language `interpretation`, and machine-readable `payload` entries.

Usage

```
morie_otis_regional_placement(...)

morie_otis_alert_state_combo(...)

morie_otis_volatility(...)

morie_otis_rc_trends(...)

morie_otis_descriptives(...)

morie_otis_dml(...)
```

Arguments

`...`	Arguments forwarded verbatim to the canonical short-named OTIS primitive (e.g. `morie_otis_rplace`, `morie_otis_astcmb`, `morie_otis_volat`, `morie_otis_rctrnd`, `morie_otis_otdesc`). See those functions for full per-primitive argument lists.

Details

Data sources: anonymized Ontario MCSCS placement records released under the Jahn v. Ontario (2020) settlement. The canonical OTIS table has 76,934 rows (FY 2022/23 – 2024/25). See `morie_otis_load` in `otis_analyze.R` for the canonical loader.

Year-lock invariant:

OTIS `UniqueIndividual_ID` (format `YYYY-XXXXX-AA`) is randomly reassigned every fiscal year and re-randomized per dataset file even within a year. The `variable_taxonomy.R` registry enforces `cross_year_safe = FALSE` for this column. Every aggregation below operates within `EndFiscalYear`; cross-year joins on the ID are forbidden by design.

References

Ontario Ministry of the Solicitor General (2025). Restrictive Confinement Detailed Dataset. https://data.ontario.ca.

Jahn v. Ontario (2020). Settlement Agreement – Inmate Data Disclosure.

Chernozhukov, V. et al. (2018). Double/debiased machine learning for treatment and structural parameters. *Econometrics Journal*, 21(1), C1-C68.

`morie_otis_psm` *Propensity-score 1:k NN matching with caliper (not yet ported).*

Description

The python implementation provides a greedy 1:k NN matcher on logit-PS with an Austin (2011) 0.2-SD caliper. In R, prefer the canonical **MatchIt** implementation (`MatchIt::matchit(method = "nearest", caliper = 0.2,std.caliper = TRUE)`); the present stub holds the python-API surface so callers can detect the rename.

Usage

```
morie_otis_psm(...)
```

Arguments

`...`	Arguments mirroring `morie_otis_aipw_ate()`.

Value

Stops with a redirect to **MatchIt**.

Examples

```
## Not run:
  morie_otis_psm(df, treatment = "d", outcome = "y",
                 covariates = "x")

## End(Not run)
```

morie_otis_psm_subclass

Propensity-score subclassification ATE (not yet ported).

Description

Rosenbaum-Rubin (1983) PS-stratification (default n_strata = 5, Cochran 1968 convention). Use `MatchIt::matchit(method = "subclass")` for an R-side equivalent.

Usage

```
morie_otis_psm_subclass(...)
```

Arguments

`...`	Arguments mirroring `morie_otis_ipw_ate()`.

Value

Stops with a redirect to **MatchIt** subclassification.

Examples

```
## Not run:
  morie_otis_psm_subclass(df, treatment = "d", outcome = "y",
                          covariates = "x")

## End(Not run)
```

morie_otis_rctrnd

Restrictive-confinement trends over time by region

Description

Per-(fiscal year, region) counts of unique individuals and total placements.

Usage

```
morie_otis_rctrnd(
  df,
  id_col = "unique_individual_id",
  year_col = "end_fiscal_year",
  region_col = "region_at_time_of_placement"
)
```

Arguments

`df`	data.frame.
`id_col`, `year_col`, `region_col`	Column names.

Value

`morie_otis_result` (the trends table is in `payload$trends`).

`morie_otis_regC_demog_contingency`

Multi-region path x Gender / Age contingency

Description

Per-person-year multi-region indicator (`regC >= 2`) cross- tabulated with Gender and Age_Category; reports chi-square + Cramer's V on each.

Usage

```
morie_otis_regC_demog_contingency(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

`morie_otis_region_alert_state_richness`

Region x alert state richness

Description

Distinct (region x alert-combo) states occupied per person-year.

Usage

```
morie_otis_region_alert_state_richness(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

morie_otis_repeat_placement_concentration

Repeat-placement concentration (Goffmanian cyclical-inmate)

Description

Expands the OTIS b09 banded counts into a per-individual placement- count vector, then reports the Gini coefficient, Hill-MLE power-law alpha, top-10\ exponential null. Reuses the `.gini_int` / `.hill_mle` helpers from `mrm_otis.R`.

Usage

```
morie_otis_repeat_placement_concentration(
  df,
  band_col = "NumberPlacements_Segregation",
  count_col = "NumberIndividuals_Segregation"
)
```

Arguments

`df`	b09 long-form data.frame.
`band_col`, `count_col`	Column names.

Value

`morie_otis_result`.

morie_otis_rplace

Regional placement matrix by age group

Description

Builds a count matrix (age x region) and the row-normalised proportion matrix of unique-individual placements for one fiscal year, optionally filtered by gender.

Usage

```
morie_otis_rplace(
  df,
  year,
  sex = NULL,
  id_col = "unique_individual_id",
  age_col = "age_category",
  region_col = "region_at_time_of_placement",
  year_col = "end_fiscal_year",
  gender_col = "gender"
)
```

Arguments

`df`	data.frame of OTIS placement records.
`year`	Integer fiscal year (e.g. `2024` for FY 2023/24).
`sex`	Optional gender filter (`"Male"` / `"Female"`); `NULL` = all.
`id_col`, `age_col`, `region_col`, `year_col`, `gender_col`	Column names.

Value

`morie_otis_result` list.

Examples

```
## Not run:
  df <- morie_otis_load()
  morie_otis_rplace(df, year = 2024)

## End(Not run)
```

`morie_otis_tps_analyze_all` *Run all OTIS-TPS overlay analyses*

Description

Run all OTIS-TPS overlay analyses

Usage

```
morie_otis_tps_analyze_all(otis_b01, tps_datasets, out_dir = NULL)
```

Arguments

`otis_b01`	OTIS b01 data.frame.
`tps_datasets`	Named list of TPS data.frames (one per category).
`out_dir`	Optional output directory for `overlay_<name>.rds`.

Value

Named list of `morie_otis_analysis_results` (`region_rollup`, `yoy_correlation`).

`morie_otis_tps_composite_overlay` *Composite overlay (alias for the YoY correlation)*

Description

Same body as `morie_otis_tps_yoy_correlation`; the alias preserves the Python entry-point name.

Usage

```
morie_otis_tps_composite_overlay(otis_b01, tps_datasets)
```

Arguments

`otis_b01`	OTIS b01 data.frame (e.g. `read.csv("b01_segregation_detailed_dataset.csv")`).
`tps_datasets`	A named list of TPS data.frames, one per category (e.g. `list(assault = <df>, robbery = <df>)`). Each data.frame must have `OCC_YEAR` or `REPORT_YEAR`.

`morie_otis_tps_overlay` *Cross-link OTIS (Ontario corrections) with TPS (Toronto police) data*

Description

R port of `morie.otis_tps_overlay`. Both feeds touch Toronto, so three overlay analyses are meaningful:

Details

- `morie_otis_tps_yoy_correlation()` – year-over-year Pearson r between OTIS Toronto-region segregation placements and TPS incident counts (per category).
- `morie_otis_tps_per_region_rollup()` – OTIS seg/RC totals per region x year, with the Toronto row flagged for overlay use.
- `morie_otis_tps_composite_overlay()` – alias for `morie_otis_tps_yoy_correlation()` (preserves the Python name, same body).

All three return a `morie_otis_analysis_result` (the shared RichResult-shaped list from `otis_all_analyze.R`).

Cross-year invariants:

- OTIS `UniqueIndividual_ID` is reassigned every fiscal year (see `variable_taxonomy.R`); the overlay therefore joins at the *year* grain, never at the person grain.

- OTIS uses fiscal-year (`EndFiscalYear`); TPS uses calendar `OCC_YEAR` or `REPORT_YEAR`. The Pearson r here is computed on the year-aligned intersection – there is a small fiscal/calendar misalignment that is documented in the interpretation but not corrected. Toronto OTIS data covers only 2023-2025, so common-year samples are necessarily small.

`morie_otis_tps_per_region_rollup`
OTIS seg/RC totals per region x year (Toronto row flagged for TPS-overlay use)

Description

OTIS seg/RC totals per region x year (Toronto row flagged for TPS-overlay use)

Usage

```
morie_otis_tps_per_region_rollup(otis_b01)
```

Arguments

`otis_b01`	OTIS b01 data.frame.

Value

A `morie_otis_analysis_result` with a year x region count matrix.

`morie_otis_tps_yoy_correlation`
Year-over-year correlation between OTIS Toronto-region segregation placements and TPS incident counts (per category)

Description

Year-over-year correlation between OTIS Toronto-region segregation placements and TPS incident counts (per category)

Usage

```
morie_otis_tps_yoy_correlation(otis_b01, tps_datasets)
```

Arguments

`otis_b01`	OTIS b01 data.frame (e.g. `read.csv("b01_segregation_detailed_dataset.csv")`).
`tps_datasets`	A named list of TPS data.frames, one per category (e.g. `list(assault = <df>, robbery = <df>)`). Each data.frame must have `OCC_YEAR` or `REPORT_YEAR`.

Value

A `morie_otis_analysis_result` with a per-category Pearson r table.

Examples

```
if (FALSE) {
  b01 <- read.csv("b01_segregation_detailed_dataset.csv")
  tps <- list(
    assault = read.csv("Assault_Open_Data.csv"),
    robbery = read.csv("Robbery_Open_Data.csv")
  )
  morie_otis_tps_yoy_correlation(b01, tps)
}
```

`morie_otis_volat` *Regional volatility / placement movement*

Description

Counts the number of distinct regions an individual was placed in within one fiscal year (union of `region_at_time_of_placement` and `region_most_recent_placement`).

Usage

```
morie_otis_volat(
  df,
  id_col = "unique_individual_id",
  year_col = "end_fiscal_year",
  regA_col = "region_at_time_of_placement",
  regB_col = "region_most_recent_placement"
)
```

Arguments

`df`	data.frame.
`id_col`, `year_col`, `regA_col`, `regB_col`	Column names.

Value

`morie_otis_result`.

`morie_otis_within_year_placement_count`
Within-year placement-count distribution

Description

Distribution of segregation placements per (individual x fiscal year) cell. Because OTIS IDs are year-locked (`YYYY-XXXXX-AA`), each cell is one anonymous person-year; cross-year readmission is not measurable.

Usage

```
morie_otis_within_year_placement_count(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

`morie_otis_within_year_region_diversity`
Within-year region diversity

Description

Distinct `Region_AtTimeOfPlacement` values per person-year.

Usage

```
morie_otis_within_year_region_diversity(df)
```

Arguments

`df`	b01 data.frame.

Value

`morie_otis_result`.

morie_paired_t_test *Paired t-test*

Description

Paired t-test

Usage

```
morie_paired_t_test(x1, x2, alternative = c("two.sided", "greater", "less"))
```

Arguments

x1	Numeric vector (before/condition 1).
x2	Numeric vector (after/condition 2).
alternative	"two.sided", "greater", or "less".

Value

Named list: t, df, p_value, ci_diff, mean_diff.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_parse_nypd_law_code *Parse an NYPD* law_code *string into its structural fields*

Description

Phase 3CCC1. NYPD law codes are space-or-zero-padded composites of a 1-4 char statute book prefix and a numeric/alpha section identifier. Examples:

Usage

```
morie_parse_nypd_law_code(law_code)
```

Arguments

law_code	Character vector of NYPD law_code strings.

Details

- "PL 1601005" -> book=PL, section=1601005 (Penal Law)
- "VTL0511000" -> book=VTL, section=0511000
- "AC 0019190" -> book=AC, section=0019190 (NYC Admin Code)
- "ABC0064A00" -> book=ABC, section=0064A00

The book prefix is extracted as the leading run of uppercase ASCII letters; the section is everything after the prefix with leading whitespace stripped. NA / empty inputs return NA fields.

Value

A data.frame with book, section columns aligned to law_code. Length-preserving.

Examples

```
morie_parse_nypd_law_code(c("PL 1601005", "AC 0019190", "ABC0064A00"))
```

---

morie_paths *Resolve standard project paths*

Description

Resolve standard project paths

Usage

```
morie_paths(project_root = NULL)
```

Arguments

project_root	Project root directory. If NULL, inferred from the current working directory.

Value

Named list of key paths.

Examples

```
tryCatch(morie_paths(),
  error = function(e) message("not inside a morie project tree")
)
```

morie_pca_dimension_reduction

PCA via SVD for dimension reduction (R parity)

Description

Wraps `stats::prcomp`.

Usage

```
morie_pca_dimension_reduction(x, n_components = NULL, seed = 0L)
```

Arguments

`x`	Numeric matrix.
`n_components`	Number of components (default min(n, p)).
`seed`	Unused for the SVD path; kept for API parity.

Value

Named list: estimate, components, explained_variance, explained_variance_ratio, singular_values, scores, n_components, n, method.

Examples

```
morie_pca_dimension_reduction(x = rnorm(50))
```

morie_pcg_filter

Phonocardiogram (PCG) bandpass filter

Description

Convenience preset wrapping `buttbp()` with the standard PCG band (25–400 Hz at 2000 Hz sampling). Removes baseline drift below 25 Hz and anti-aliased high-frequency noise above 400 Hz.

Usage

```
morie_pcg_filter(x, fs = 2000, low = 25, high = 400)
```

Arguments

`x`	Numeric vector (PCG signal).
`fs`	Sampling frequency (Hz, default 2000).
`low`	Lower cutoff (Hz, default 25).
`high`	Upper cutoff (Hz, default 400).

Value

List with filtered signal (see `buttbp()`).

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  x <- rnorm(2000) # 1 second of white-noise PCG-like input
  y <- morie_pcg_filter(x)
  length(y$filtered)
}
```

---

morie_penalized_regression

Elastic-net regression via coordinate descent (base R)

Description

Uses glmnet if available; otherwise the base-R coordinate-descent fallback. Both solve:

Usage

```
morie_penalized_regression(
  x,
  y,
  alpha = 0.5,
  lam = 1,
  max_iter = 1000,
  tol = 1e-06
)
```

Arguments

`x`	(n x p) predictor matrix.
`y`	Numeric response.
`alpha`	0 (ridge) to 1 (LASSO).
`lam`	Penalty strength.
`max_iter`, `tol`	Convergence controls.

Details

min 1/(2n) ||y - X beta||^2 + lam (alpha ||beta||_1 + (1-alpha)/2 ||beta||_2^2).

Value

list(estimate, beta, intercept, se, alpha, lam, n_iter, n, p, method).

References

Friedman, Hastie & Tibshirani (2010); Montesinos Lopez Ch 6.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_percentile_modified_rank` *Percentile-modified rank (Gastwirth) two-sample test (Gibbons Ch 8.3.3)*

Description

Trims central ranks; only tail ranks contribute. Score: a_i = max(R_i - (1-q)(N+1), 0) - max(q(N+1) - R_i, 0)

Usage

```
morie_percentile_modified_rank(x, y, q = 0.25)
```

Arguments

`x, y`	Numeric vectors.
`q`	Tail fraction in (0, 0.5). Default 0.25.

Value

Named list: statistic, p_value, z, n, m, q.

Examples

```
morie_percentile_modified_rank(x = rnorm(50), y = rnorm(50))
```

`morie_performance_check_collinearity`
Collinearity / VIF check via **performance**

Description

Thin extender over `performance::check_collinearity` for the variance-inflation-factor diagnostic.

Usage

```
morie_performance_check_collinearity(model, ...)
```

Arguments

`model`	A fitted model object supported by **performance**.
`...`	Further arguments forwarded to `performance::check_collinearity` (e.g. `ci`, `verbose`).

Value

A list with `$method = "performance::check_collinearity"` and `$raw` (the upstream VIF data frame).

`morie_performance_check_model`
Regression-diagnostic plots via **performance**

Description

Thin extender over `performance::check_model` that returns the diagnostic-plot grob list for a fitted model.

Usage

```
morie_performance_check_model(model, ...)
```

Arguments

`model`	A fitted model object supported by **insight** / **performance**.
`...`	Further arguments forwarded to `performance::check_model` (e.g. `check`, `panel`, `theme`).

Value

A list with `$method = "performance::check_model"` and `$raw` (the upstream `check_model` object).

morie_performance_check_outliers *Outlier check via* **performance**

Description

Thin extender over `performance::check_outliers` for composite outlier detection on a fitted model or numeric data.

Usage

```
morie_performance_check_outliers(x, ...)
```

Arguments

`x`	A fitted model object or numeric data frame supported by `performance::check_outliers`.
`...`	Further arguments forwarded to `performance::check_outliers` (e.g. `method`, `threshold`).

Value

A list with `$method = "performance::check_outliers"` and `$raw` (the upstream outlier-check object).

morie_performance_r2 *R-squared family via* **performance**

Description

Thin extender over `performance::r2` returning the appropriate R-squared (Nakagawa, McFadden, Tjur, ...) for a supported fitted model.

Usage

```
morie_performance_r2(model, ...)
```

Arguments

`model`	A fitted model object supported by **performance**.
`...`	Further arguments forwarded to `performance::r2` (e.g. `tolerance`, `ci`, `verbose`).

Value

A list with `$method = "performance::r2"` and `$raw` (the upstream R-squared object).

`morie_point_biserial_r`
Point-biserial correlation

Description

Point-biserial correlation

Usage

```
morie_point_biserial_r(binary_var, continuous_var)
```

Arguments

`binary_var`	Binary numeric vector (0/1).
`continuous_var`	Continuous numeric vector.

Value

Named list: `r`, `p_value`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_polynomial_regression`
Polynomial regression (R parity)

Description

Polynomial feature expansion + OLS via `stats::poly` + `stats::lm`. Uses raw (not orthogonal) polynomials for parity with scikit-learn's PolynomialFeatures.

Usage

```
morie_polynomial_regression(x, y, degree = 2L)
```

Arguments

`x`	Numeric vector or matrix.
`y`	Numeric response.
`degree`	Polynomial degree.

Value

Named list: estimate, se, feature_names, degree, n, method.

Examples

```
morie_polynomial_regression(x = rnorm(50), y = rnorm(50))
```

---

morie_power_prop_test

Power for a two-proportion z-test

Description

Mirrors R's `power.prop.test()`.

Usage

```
morie_power_prop_test(
  n = NULL,
  p1 = NULL,
  p2 = NULL,
  sig_level = 0.05,
  power = NULL,
  alternative = c("two.sided", "one.sided")
)
```

Arguments

`n`	Sample size per group.
`p1`	Proportion in group 1.
`p2`	Proportion in group 2.
`sig_level`	Type I error rate.
`power`	Desired power.
`alternative`	`"two.sided"` or `"one.sided"`.

Value

Result of `stats::power.prop.test()`.

Examples

```
morie_power_prop_test(p1 = 0.30, p2 = 0.20, power = 0.80)
```

`morie_power_t_test` *Power for a two-sample t-test*

Description

Solve for any missing parameter (`n`, `delta`, `sd`, `sig.level`, or `power`). Mirrors R's `power.t.test()`.

Usage

```
morie_power_t_test(
  n = NULL,
  delta = NULL,
  sd = 1,
  sig_level = 0.05,
  power = NULL,
  alternative = c("two.sided", "one.sided"),
  type = c("two.sample", "one.sample", "paired")
)
```

Arguments

`n`	Sample size per group (NULL to solve for it).
`delta`	Effect size (difference in means).
`sd`	Standard deviation (pooled).
`sig_level`	Type I error rate (alpha).
`power`	Desired power (1 - beta).
`alternative`	`"two.sided"` or `"one.sided"`.
`type`	`"two.sample"`, `"one.sample"`, or `"paired"`.

Value

Result of `stats::power.t.test()`.

Examples

```
morie_power_t_test(n = NULL, delta = 0.5, power = 0.80)
```

morie_ppcor_partial *Partial correlation via* **ppcor**

Description

Thin extender over `ppcor::pcor` (matrix-wise) or `ppcor::pcor.test` (when `y` and `z` are supplied) for partial correlations controlling for one or more variables.

Usage

```
morie_ppcor_partial(x, y = NULL, z = NULL, method = "pearson", ...)
```

Arguments

`x`	Numeric vector, matrix, or data frame. When `y` and `z` are `NULL`, the full pairwise partial correlation matrix is returned via `ppcor::pcor`.
`y`	Optional second numeric vector.
`z`	Optional numeric vector / matrix of control variables.
`method`	Correlation method (`"pearson"`, `"spearman"`, or `"kendall"`), forwarded to ppcor.
`...`	Further arguments forwarded to the upstream function.

Value

A list with `$method` (qualified upstream name) and `$raw` (the upstream return object).

morie_ppcor_semipartial *Semi-partial correlation via* **ppcor**

Description

Thin extender over `ppcor::spcor` (matrix-wise) or `ppcor::spcor.test` (when `y` and `z` are supplied) for semi-partial (part) correlations.

Usage

```
morie_ppcor_semipartial(x, y = NULL, z = NULL, method = "pearson", ...)
```

Arguments

`x`	Numeric vector, matrix, or data frame. When `y` and `z` are `NULL`, the full pairwise semi-partial correlation matrix is returned via `ppcor::spcor`.
`y`	Optional second numeric vector.
`z`	Optional numeric vector / matrix of control variables.
`method`	Correlation method (`"pearson"`, `"spearman"`, or `"kendall"`), forwarded to ppcor.
`...`	Further arguments forwarded to the upstream function.

Value

A list with `$method` (qualified upstream name) and `$raw` (the upstream return object).

`morie_pps_sample` *Probability proportional to size (PPS) sampling*

Description

Probability proportional to size (PPS) sampling

Usage

```
morie_pps_sample(df, size_col, n, seed = 42L, replace = FALSE)
```

Arguments

`df`	A data frame.
`size_col`	Name of the size measure column.
`n`	Number of units to select.
`seed`	Random seed.
`replace`	Logical; `FALSE` (default) uses PPS-without-replacement (Madow systematic-like), matching `morie.sampling.pps_sample` since 2026-05-22. `TRUE` reverts to the legacy Hansen-Hurwitz with-replacement scheme.

Value

Data frame of selected units with `.weight` (Hansen-Hurwitz weights).

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_prediction_accuracy

Genomic-prediction accuracy metrics

Description

Reports Pearson r, Spearman rho, MSE/MSPE, RMSE, R^2, calibration slope and intercept.

Usage

```
morie_prediction_accuracy(y_true, y_pred)
```

Arguments

y_true	Numeric observed.
y_pred	Numeric predicted.

Value

list(estimate (Pearson r), pearson_r, morie_spearman_rho, mse, mspe, rmse, r2, slope, intercept, n, method).

References

Montesinos Lopez Ch 2.

Examples

```
morie_prediction_accuracy(y_true = rbinom(50, 1, 0.5), y_pred = rbinom(50, 1, 0.5))
```

morie_predpol_aggregate_areas

Aggregate per-record predictive-policing data to one row per area

Description

Aggregate per-record predictive-policing data to one row per area

Usage

```
morie_predpol_aggregate_areas(
  area,
  risk,
  outcome,
  group = NULL,
  population = NULL
)
```

Arguments

`area`	Area identifier for each record.
`risk`	Predicted risk score for each record.
`outcome`	Realised-outcome indicator/count for each record.
`group`	Optional protected attribute per record; the per-area majority value becomes that area's group label.
`population`	Optional area population: a named numeric vector (`area -> population`) or a per-record vector. When given, the outcome rate is per 10,000 inhabitants; otherwise it is the mean outcome per record.

Value

A named list: `areas`, `mean_risk`, `outcome_rate`, `group`, `n_records`.

Examples

```
agg <- morie_predpol_aggregate_areas(
  area = c("a", "a", "b", "b"), risk = c(10, 20, 30, 40),
  outcome = c(1, 0, 1, 1)
)
agg$mean_risk # 15 35
agg$outcome_rate # 0.5 1.0
```

`morie_predpol_calibration_audit`

Audit whether an algorithm's area risk ranking matches realised outcomes

---

Description

Ranks areas by predicted risk and by realised outcome rate (rank 1 = highest), forms `rank_gap = outcome_rank - risk_rank` per area (positive = over-predicted), and averages the gap within each group. A Spearman correlation summarises overall calibration.

Usage

```
morie_predpol_calibration_audit(areas, mean_risk, outcome_rate, group)
```

Arguments

`areas`	Area identifiers (one per area).
`mean_risk`	Mean predicted risk per area.
`outcome_rate`	Realised outcome rate per area.
`group`	Majority protected-attribute label per area.

Value

A named list: `value` (worst per-group mean gap), `spearman`, `spearman_pvalue`, `group_rank_gap`, `worst_group`, `rank_gap`, `warnings`, `interpretation`.

Examples

```
res <- morie_predpol_calibration_audit(
  areas = c("d1", "d2", "d3", "d4", "d5", "d6"),
  mean_risk = c(90, 80, 70, 30, 20, 10),
  outcome_rate = c(10, 20, 30, 70, 80, 90),
  group = c("X", "X", "X", "Y", "Y", "Y")
)
res$group_rank_gap$X # 3  (group X over-predicted)
res$spearman # -1 (perfectly miscalibrated)
```

`morie_predpol_score_disparity`

Descriptive disparity in a risk score across groups

Description

Reports per-group n / mean / median / sd, a one-way ANOVA for whether group membership relates to the score, and each group's mean-score gap from a reference group. A significant gap is not itself proof of bias; pair this with `morie_predpol_calibration_audit()`.

Usage

```
morie_predpol_score_disparity(score, group, reference = NULL)
```

Arguments

`score`	Continuous risk score, one per individual.
`group`	Protected attribute, one per individual.
`reference`	Reference group for the gaps; defaults to the lowest-scoring group.

Value

A named list: `value` (mean-score spread), `spread`, `group_means`, `gaps`, `anova_f`, `anova_pvalue`, `significant`, `reference`, `warnings`, `interpretation`.

Examples

```
res <- morie_predpol_score_disparity(
  score = c(9, 10, 11, 19, 20, 21),
  group = c("A", "A", "A", "B", "B", "B")
)
res$value # 10  (group means 10 and 20)
res$significant # TRUE
```

`morie_predpol_temporal_audit`
Audit how disparity metrics move over time and across cities

Description

For every `(city, period)` cell the four disparity metrics are computed; per city the audit then reports the mean of each metric, the count of periods with DIR above 1, and the DIR temporal range (max minus min) — the headline measure of instability.

Usage

```
morie_predpol_temporal_audit(
  period,
  city,
  y_pred,
  group,
  privileged = NULL,
  favorable = 1
)
```

Arguments

`period`	Time-period label for each record (e.g. `"2019-03"`).
`city`	City label for each record.
`y_pred`	The decision/assignment for each record.
`group`	Protected attribute for each record.
`privileged`	Reference group; inferred globally from the pooled data when `NULL` so every cell uses the same reference.
`favorable`	Value of `y_pred` counted as favourable (default `1`).

Value

A named list: `value` (worst per-city DIR range), `worst_dir_range`, `cross_city_dir_spread`, `per_city`, `cells`, `privileged`, `warnings`, `interpretation`.

Examples

```
period <- c(rep("p1", 10), rep("p2", 10))
city <- rep("A", 20)
pred <- rep(c(1, 1, 1, 1, 1, 1, 1, 1, 0, 0), 2)
grp <- rep(c(rep("X", 5), rep("Y", 5)), 2)
res <- morie_predpol_temporal_audit(period, city, pred, grp, privileged = "X")
res$per_city$A$dir_range # 0 - disparity is stable across periods
```

morie_profile_dataset

Profile a data.frame: per-column types, missingness, summary statistics

Description

Mirrors the Python `morie.profile_dataset()`. Returns a list of per-column profiles plus dataset-level metadata.

Usage

```
morie_profile_dataset(df)
```

Arguments

`df`	A `data.frame`.

Value

A list with components:

`n_rows`, `n_cols` Dataset dimensions.

`columns` A named list, one entry per column, each containing `name`, `dtype`, `measurement_level`, `n_missing`, `n_unique`, and (for numeric columns) `mean`, `sd`, `min`, `max`, `q25`, `q50`, `q75`.

Examples

```
p <- morie_profile_dataset(iris)
p$columns$Species
p$columns$Sepal.Length
```

morie_prophet_components

Prophet-style additive decomposition (linear trend + Fourier seasonality)

Description

Prophet-style additive decomposition (linear trend + Fourier seasonality)

Usage

```
morie_prophet_components(x, period = 12)
```

Arguments

`x`	Numeric univariate series.
`period`	Seasonal period. Default 12.

Value

Named list with `trend`, `seasonal`, `residual`, `slope`, `intercept`, `fourier_terms`, `period`, `n`, `method`.

Examples

```
morie_prophet_components(x = rnorm(50))
```

`morie_proportion_ci` *Wilson score confidence interval for a proportion*

Description

Wilson score confidence interval for a proportion

Usage

```
morie_proportion_ci(
  successes,
  n,
  alpha = 0.05,
  method = c("wilson", "exact", "wald")
)
```

Arguments

`successes`	Number of successes.
`n`	Total observations.
`alpha`	Significance level (default 0.05 -> 95% CI).
`method`	`"wilson"` (default), `"exact"` (Clopper-Pearson), or `"wald"`.

Value

Named list: `p_hat`, `ci_lower`, `ci_upper`.

Examples

```
morie_proportion_ci(35, 100)
```

morie_psymet_alpha *Cronbach's coefficient alpha with Feldt CI*

Description

Hand-rolled base-R implementation. When the `psych` package is installed, results agree with `psych::alpha()$total` to numerical precision.

Usage

```
morie_psymet_alpha(data, ci = 0.95)
```

Arguments

`data`	Numeric matrix or data.frame: items as columns, respondents as rows.
`ci`	Confidence level (default 0.95).

Value

A list with components `raw`, `std`, `avgr`, `k`, `n`, `ci_lo`, `ci_hi`.

morie_psymet_alphadel *Alpha if item deleted*

Description

Alpha if item deleted

Usage

```
morie_psymet_alphadel(data)
```

Arguments

`data`	Numeric matrix / data.frame.

Value

data.frame with `item`, `adel`.

`morie_psymet_ave` *Average variance extracted (AVE) from factor loadings. Mean(lambda^2).*

Description

Average variance extracted (AVE) from factor loadings. Mean(lambda^2).

Usage

```
morie_psymet_ave(loads)
```

Arguments

`loads`	Numeric vector of standardised factor loadings (lambda).

`morie_psymet_bartlett` *Bartlett's test of sphericity.*

Description

Bartlett's test of sphericity.

Usage

```
morie_psymet_bartlett(data)
```

Arguments

`data`	Numeric matrix or data.frame of items.

Value

list with `chisq`, `df`, `pval`.

morie_psymet_cr — *Composite reliability from standardized factor loadings. CR = (sum lambda)^2 / ((sum lambda)^2 + sum(1 - lambda^2))*

Description

Composite reliability from standardized factor loadings. CR = (sum lambda)^2 / ((sum lambda)^2 + sum(1 - lambda^2))

Usage

```
morie_psymet_cr(loads)
```

Arguments

`loads`	Numeric vector of standardised factor loadings (lambda).

morie_psymet_discrimination — *Item discrimination (D-statistic).*

Description

Upper/lower groups by total score (default 27% per Kelley).

Usage

```
morie_psymet_discrimination(data, pct = 0.27)
```

Arguments

`data`	Numeric matrix or data.frame of items.
`pct`	Numeric in (0, 0.5); proportion for the upper/lower group split (default 0.27, the Kelley-Cureton rule).

Value

data.frame with `item`, `d`.

morie_psymet_itemtotal *Corrected item-total correlations*

Description

Corrected item-total correlations

Usage

```
morie_psymet_itemtotal(data)
```

Arguments

`data`	Numeric matrix / data.frame.

Value

data.frame with columns `item`, `r_total`, `r_corr`.

morie_psymet_kmo *Kaiser-Meyer-Olkin sampling adequacy.*

Description

Delegates to `psych::KMO()` when available; otherwise computes from the partial-correlation anti-image matrix using base R.

Usage

```
morie_psymet_kmo(data)
```

Arguments

`data`	Numeric matrix or data.frame of items.

Value

list with `msa` (overall) and named numeric vector `items`.

morie_psymet_omega *McDonald's omega total and hierarchical*

Description

Delegates to `psych::omega()` when available; otherwise uses a single-factor principal-axis approximation.

Usage

```
morie_psymet_omega(data, nf = 1)
```

Arguments

`data`	Numeric matrix / data.frame of items.
`nf`	Number of factors (default 1).

Value

list with `total`, `hier`, `alpha`, `nf`, `expvar`.

morie_psymet_parallel *Horn's parallel analysis – suggested number of factors.*

Description

Delegates to `psych::fa.parallel()` when available; otherwise compares observed eigenvalues to the 95th percentile of random-data eigenvalues.

Usage

```
morie_psymet_parallel(data, nsim = 100, seed = 42)
```

Arguments

`data`	Numeric matrix or data.frame of items.
`nsim`	Integer; number of simulated random datasets (default 100).
`seed`	Integer; RNG seed for reproducibility.

`morie_psymet_splithalf`
Spearman-Brown split-half reliability.

Description

Spearman-Brown split-half reliability.

Usage

```
morie_psymet_splithalf(data, method = c("first_last", "odd_even"))
```

Arguments

`data`	Numeric matrix or data.frame of items.
`method`	"first_last" or "odd_even".

`morie_quantile_reg` *Quantile regression via* **quantreg**

Description

Thin extender over `quantreg::rq` for Koenker-Bassett quantile regression at one or more conditional quantiles (Koenker & Bassett, 1978; Koenker, 2005).

Usage

```
morie_quantile_reg(formula, tau = 0.5, data, ...)
```

Arguments

`formula`	A model formula of the form `y ~ x1 + x2 + ...`.
`tau`	Numeric scalar or vector in `(0, 1)`; the conditional quantile(s) at which to fit the regression (default `0.5`, the median).
`data`	A data frame containing the variables in `formula`.
`...`	Further arguments forwarded to `quantreg::rq` (e.g. `subset`, `weights`, `na.action`, `method`, `model`, `contrasts`).

Value

A list with `$method = "quantreg::rq"` and `$raw` (an `rq` / `rqs` object with the fitted coefficients at each `tau`).

Examples

```
## Not run:
  if (requireNamespace("quantreg", quietly = TRUE)) {
    set.seed(1)
    n  <- 100
    df <- data.frame(x = stats::rnorm(n))
    df$y <- 1 + 2 * df$x + stats::rnorm(n)
    morie_quantile_reg(y ~ x, tau = c(0.25, 0.5, 0.75), data = df)
  }

## End(Not run)
```

---

morie_random_forest_ensemble

Random Forest ensemble (R parity)

Description

Wraps `randomForest::randomForest`. Auto-detects task from y (factor / integer-like -> classification, otherwise regression).

Usage

```
morie_random_forest_ensemble(
  x,
  y,
  n_estimators = 100L,
  max_depth = NULL,
  task = "auto",
  seed = 0L,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric predictor matrix.
`y`	Response.
`n_estimators`	Number of trees.
`max_depth`	Max tree depth (NULL -> unrestricted).
`task`	"auto", "classification", or "regression".
`seed`	RNG seed.
`deterministic_seed`	Integer or NULL. If supplied, the RNG state is derived from the SHA-keyed `morie_det_rng()` so Py<->R streams agree on the canonical fixture. When `NULL` (default), behaviour is unchanged: `seed` drives `set.seed()` directly.

Value

Named list: estimate, train_score, oob_score, feature_importances, n_estimators, task, n, method.

Examples

```
morie_random_forest_ensemble(x = rnorm(50), y = rnorm(50))
```

---

morie_random_forest_genomic

Random-forest genomic predictor

Description

Uses randomForest if available; otherwise a base-R bagged-tree fallback (regression CART approximation).

Usage

```
morie_random_forest_genomic(
  x,
  y,
  markers,
  n_trees = 100,
  max_depth = 10,
  min_samples = 2,
  mtry = NULL,
  seed = 0
)
```

Arguments

x	Optional fixed features.
y	Numeric response.
markers	Genotype matrix (n x m).
n_trees	Number of trees.
max_depth	Max depth (fallback only).
min_samples	Min samples per node.
mtry	Features sampled per split (default sqrt(p)).
seed	Seed.

Value

list(estimate, y_hat, oob_score, feature_importance, se, n, method).

References

Breiman (2001); Montesinos Lopez Ch 8.

Examples

```
morie_random_forest_genomic(
  x = rnorm(50), y = rnorm(50),
  markers = matrix(sample(0:2, 200, TRUE), 50, 4)
)
```

---

morie_random_search_cv

Random search hyperparameter optimisation (R parity)

Description

Uses `caret::train` with `search = "random"`.

Usage

```
morie_random_search_cv(
  x,
  y,
  method = NULL,
  n_iter = 20L,
  cv = 5L,
  task = "auto",
  seed = 0L,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric predictor matrix.
`y`	Response.
`method`	caret method id (default by task).
`n_iter`	Number of random draws.
`cv`	CV folds.
`task`	"auto" / "classification" / "regression".
`seed`	RNG seed.
`deterministic_seed`	Integer or NULL. If supplied, the RNG state is derived from the SHA-keyed `morie_det_rng()` so Py<->R streams agree on the canonical fixture. When `NULL` (default), behaviour is unchanged: `seed` drives `set.seed()` directly.

Value

Named list: estimate, best_params, best_score, sampled_params, sampled_scores, n_iter, task, n, method.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

---

morie_randtests_bartels

Bartels rank test via **randtests**

Description

Thin extender over `randtests::bartels.rank.test` for the Bartels rank von-Neumann test of randomness in a numeric sequence.

Usage

```
morie_randtests_bartels(x, ...)
```

Arguments

`x`	A numeric vector.
`...`	Further arguments forwarded to `randtests::bartels.rank.test` (e.g. `alternative`, `pvalue`).

Value

A list with `$method = "randtests::bartels.rank.test"` and `$raw` (an `htest` object).

---

morie_randtests_runs *Wald-Wolfowitz runs test via* **randtests**

Description

Thin extender over `randtests::runs.test` for a non-parametric test of randomness in a numeric sequence.

Usage

```
morie_randtests_runs(x, ...)
```

Arguments

`x`	A numeric vector.
`...`	Further arguments forwarded to `randtests::runs.test` (e.g. `alternative`, `threshold`, `pvalue`, `plot`).

Value

A list with `$method = "randtests::runs.test"` and `$raw` (an `htest` object).

`morie_randtests_turning_point`

Turning-point test via **randtests**

Description

Thin extender over `randtests::turning.point.test` for the classical turning-point test of randomness in a numeric sequence.

Usage

```
morie_randtests_turning_point(x, ...)
```

Arguments

`x`	A numeric vector.
`...`	Further arguments forwarded to `randtests::turning.point.test` (e.g. `alternative`).

Value

A list with `$method = "randtests::turning.point.test"` and `$raw` (an `htest` object).

`morie_rank_based_test`

Mann's rank test for randomness (Gibbons Ch 3.5)

Description

Kendall tau between the observation and its time index t = 1..n. Tests H0: no monotone trend.

Usage

```
morie_rank_based_test(x)
```

Arguments

`x`	Numeric vector of sequential observations.

Value

Named list: statistic (tau), p_value, n, inversions, z.

Examples

```
morie_rank_based_test(x = rnorm(50))
```

`morie_rank_order_statistics`

Signed ranks of paired differences (Gibbons Ch 5.5)

Description

Signed ranks R_i^+ = sign(D_i) * rank(|D_i|) used by Wilcoxon signed-rank.

Usage

```
morie_rank_order_statistics(x, mu0 = 0)
```

Arguments

`x`	Numeric vector of differences (or values; mu0 is subtracted).
`mu0`	Hypothesised median (default 0).

Value

Named list: signed_ranks, abs_ranks, W_plus, W_minus, n_nonzero, n.

Examples

```
morie_rank_order_statistics(x = rnorm(50))
```

morie_rank_placements

Rank placements of Y among X order statistics (Gibbons Ch 2.11.3)

Description

For each Y_j: placement P_j = number of X_i less than Y_j. Their sum is the Mann-Whitney U statistic for Y vs X.

Usage

```
morie_rank_placements(x, y)
```

Arguments

`x`, `y`	Numeric vectors.

Value

Named list: placements, ranks_y, U_y, E_U, Var_U, m, n.

Examples

```
morie_rank_placements(x = rnorm(50), y = rnorm(50))
```

morie_rdd_bandwidth_cct

Calonico-Cattaneo-Titiunik (CCT) MSE-optimal bandwidth

Description

Calonico-Cattaneo-Titiunik (CCT) MSE-optimal bandwidth

Usage

```
morie_rdd_bandwidth_cct(x, y, cutoff = 0, kernel = "triangular", p = 1)
```

Arguments

`x`	Numeric vector of running-variable values (used by bandwidth selectors + density tests that don't take a `data.frame`).
`y`	Numeric vector of outcome values aligned with `x`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.

`morie_rdd_bandwidth_ik` *Imbens-Kalyanaraman (IK) MSE-optimal bandwidth*

Description

Dispatches to `rdrobust::rdbwselect(bwselect = "mserd")` which implements the modern IK-equivalent CCT MSE-optimal rule.

Usage

```
morie_rdd_bandwidth_ik(x, y, cutoff = 0, kernel = "triangular")
```

Arguments

`x`	Numeric vector of running-variable values (used by bandwidth selectors + density tests that don't take a `data.frame`).
`y`	Numeric vector of outcome values aligned with `x`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.

`morie_rdd_bandwidth_rot` *Rule-of-thumb (ROT) bandwidth – Silverman-style on running variable*

Description

Rule-of-thumb (ROT) bandwidth – Silverman-style on running variable

Usage

```
morie_rdd_bandwidth_rot(x, y, cutoff = 0)
```

Arguments

`x`	Numeric vector of running-variable values (used by bandwidth selectors + density tests that don't take a `data.frame`).
`y`	Numeric vector of outcome values aligned with `x`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).

morie_rdd_bandwidth_sensitivity

Bandwidth sensitivity sweep

Description

Bandwidth sensitivity sweep

Usage

```
morie_rdd_bandwidth_sensitivity(
  data,
  outcome,
  running,
  cutoff = 0,
  bandwidth_range = NULL,
  p = 1,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`bandwidth_range`	Numeric vector of candidate bandwidths used by the sensitivity analysis.
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

morie_rdd_bias_corrected

CCT bias-corrected, robust-SE RDD inference

Description

CCT bias-corrected, robust-SE RDD inference

Usage

```
morie_rdd_bias_corrected(
  data,
  outcome,
  running,
  cutoff = 0,
  bandwidth = NULL,
  rho = 1,
  p = 1,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`rho`	Bandwidth ratio for bias correction (Calonico, Cattaneo & Titiunik 2014); default `1` (same bandwidth).
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

`morie_rdd_cattaneo_density` *Cattaneo-Jansson-Ma (2020) local-polynomial density test*

Description

Cattaneo-Jansson-Ma (2020) local-polynomial density test

Usage

```
morie_rdd_cattaneo_density(
  x,
  cutoff = 0,
  p = 2,
  kernel = "triangular",
  bandwidth = NULL
)
```

Arguments

x	Numeric vector of running-variable values (used by bandwidth selectors + density tests that don't take a `data.frame`).
cutoff	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
p	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
kernel	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
bandwidth	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.

---

morie_rdd_covariate_balance

Covariate balance at the cutoff

Description

Runs a sharp-RDD null test on each covariate.

Usage

```
morie_rdd_covariate_balance(
  data,
  running,
  covariates,
  cutoff = 0,
  bandwidth = NULL,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

data	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
running	Character; column name of the running (forcing) variable in `data`.
covariates	Character vector of column names whose balance at the cutoff is checked.
cutoff	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
bandwidth	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
kernel	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
alpha	Significance level (default `0.05`).

`morie_rdd_density_test`
McCrary-style RD density test via **rddensity**

Description

Thin extender over `rddensity::rddensity` for the manipulation / discontinuity-in-density test of Cattaneo, Jansson & Ma (2020), the modern replacement for the original McCrary (2008) test.

Usage

```
morie_rdd_density_test(X, cutoff = 0, ...)
```

Arguments

`X`	Numeric vector of the running / forcing variable.
`cutoff`	Numeric scalar; the threshold value of `X` defining the discontinuity (default `0`).
`...`	Further arguments forwarded to `rddensity::rddensity` (e.g. `p`, `q`, `kernel`, `vce`, `h`).

Value

A list with `$method = "rddensity::rddensity"` and `$raw` (an `rddensity` object containing the estimates and the manipulation test statistic).

Examples

```
## Not run:
  if (requireNamespace("rddensity", quietly = TRUE)) {
    set.seed(1)
    x <- c(rnorm(500, -0.2), rnorm(500, 0.2))
    morie_rdd_density_test(x, cutoff = 0)
  }

## End(Not run)
```

`morie_rdd_discrete` *RDD with discrete running variable*

Description

RDD with discrete running variable

Usage

```
morie_rdd_discrete(
  data,
  outcome,
  running,
  cutoff = 0,
  bandwidth = NULL,
  p = 0,
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
`alpha`	Significance level (default `0.05`).

`morie_rdd_donut` *Donut-hole RDD*

Description

Donut-hole RDD

Usage

```
morie_rdd_donut(
  data,
  outcome,
  running,
  cutoff = 0,
  donut = 0,
  bandwidth = NULL,
  p = 1,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`donut`	Numeric; symmetric window around the cutoff to drop in a donut-RDD robustness check (default `0`).
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

`morie_rdd_fuzzy` *Fuzzy RDD treatment effect via instrumented Wald ratio*

Description

Fuzzy RDD treatment effect via instrumented Wald ratio

Usage

```
morie_rdd_fuzzy(
  data,
  outcome,
  running,
  treatment,
  cutoff = 0,
  bandwidth = NULL,
  p = 1,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.

`treatment`	Character; column name of the treatment-receipt variable (fuzzy designs).
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

`morie_rdd_geographic` *Geographic / boundary RDD on a signed distance*

Description

Geographic / boundary RDD on a signed distance

Usage

```
morie_rdd_geographic(
  data,
  outcome,
  distance_to_boundary,
  side,
  bandwidth = NULL,
  p = 1,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`distance_to_boundary`	Character; column name of the signed distance to the geographic boundary in `data`.
`side`	Character; column name encoding the treatment side (e.g. `"left"`/`"right"`).
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`p`	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

morie_rdd_kernels *RDD kernel functions*

Description

Vectorised kernel functions on the support |u| <= 1 (Gaussian is on the real line). Used by RDD local-linear estimators and friends for kernel weighting around the cutoff.

Usage

```
morie_rdd_kernel_triangular(u)

morie_rdd_kernel_epanechnikov(u)

morie_rdd_kernel_uniform(u)

morie_rdd_kernel_gaussian(u)
```

Arguments

u	Numeric vector of standardised distances from the cutoff (i.e. $(x - c)/h$).

Details

- `morie_rdd_kernel_triangular`: $K(u) = \max(1 - |u|, 0)$
- `morie_rdd_kernel_epanechnikov`: $K(u) = (3/4)(1 - u^2)$ on |u| <= 1
- `morie_rdd_kernel_uniform`: $K(u) = 1/2$ on |u| <= 1
- `morie_rdd_kernel_gaussian`: $K(u) = \phi(u)$, the standard normal density

Value

Numeric vector of kernel weights, same length as `u`.

morie_rdd_kink *Regression kink design – slope discontinuity at the cutoff*

Description

Regression kink design – slope discontinuity at the cutoff

Usage

```
morie_rdd_kink(
  data,
  outcome,
  running,
  cutoff = 0,
  bandwidth = NULL,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

`morie_rdd_local_polynomial` *Local polynomial regression at user-supplied evaluation points*

Description

Local polynomial regression at user-supplied evaluation points

Usage

```
morie_rdd_local_polynomial(x, y, eval_points, h, p = 1, kernel = "triangular")
```

Arguments

`x`	Running variable (numeric).
`y`	Outcome (numeric).
`eval_points`	Points at which to evaluate the fit.
`h`	Bandwidth.
`p`	Polynomial order (default 1, i.e. local linear).
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.

Value

A data frame of fitted values and standard errors.

`morie_rdd_local_randinf` *Local-randomisation RDD inference via* **rdlocrand**

Description

Thin extender over `rdlocrand::rdrandinf` for finite-sample randomisation-inference around the cutoff window (Cattaneo, Frandsen & Titiunik, 2015; Cattaneo, Titiunik & Vazquez-Bare, 2016).

Usage

```
morie_rdd_local_randinf(Y, R, wl, wr, ...)
```

Arguments

`Y`	Numeric outcome vector.
`R`	Numeric running / forcing variable.
`wl`	Numeric scalar; left edge of the randomisation window.
`wr`	Numeric scalar; right edge of the randomisation window.
`...`	Further arguments forwarded to `rdlocrand::rdrandinf` (e.g. `statistic`, `p`, `nulltau`, `reps`, `seed`).

Value

A list with `$method = "rdlocrand::rdrandinf"` and `$raw` (an `rdrandinf` object with the randomisation-inference p-values and the observed statistic).

Examples

```
## Not run:
  if (requireNamespace("rdlocrand", quietly = TRUE)) {
    set.seed(1)
    R <- runif(200, -1, 1)
    Y <- 0.5 * R + (R >= 0) * 0.3 + rnorm(200, sd = 0.5)
    morie_rdd_local_randinf(Y, R, wl = -0.1, wr = 0.1)
  }

## End(Not run)
```

morie_rdd_local_randomisation *Local-randomisation RDD via permutation in a fixed window*

Description

Local-randomisation RDD via permutation in a fixed window

Usage

```
morie_rdd_local_randomisation(
  data,
  outcome,
  running,
  cutoff = 0,
  window = 1,
  n_permutations = 1000,
  seed = 42,
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`window`	Numeric; half-width of the local randomisation window.
`n_permutations`	Integer; permutation count for the randomisation-based inference.
`seed`	Integer; RNG seed for permutation / bootstrap routines.
`alpha`	Significance level (default `0.05`).

morie_rdd_mccrary *McCrary (2008) density manipulation test*

Description

McCrary (2008) density manipulation test

Usage

```
morie_rdd_mccrary(x, cutoff = 0, n_bins = 50, bandwidth = NULL)
```

Arguments

`x`	Numeric vector of running-variable values (used by bandwidth selectors + density tests that don't take a `data.frame`).
`cutoff`	Numeric scalar; the threshold on `running`. Default `0` (the canonical normalisation).
`n_bins`	Integer; bin count for histogram-based density tests and binned-plot reductions.
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.

`morie_rdd_placebo_cutoff` *Placebo cutoff falsification test*

Description

Placebo cutoff falsification test

Usage

```
morie_rdd_placebo_cutoff(
  data,
  outcome,
  running,
  true_cutoff,
  placebo_cutoffs,
  bandwidth = NULL,
  p = 1,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`data`	A `data.frame` holding the outcome, running variable, treatment, and any covariates referenced by name.
`outcome`	Character; column name of the response variable in `data`.
`running`	Character; column name of the running (forcing) variable in `data`.
`true_cutoff`	Numeric; the actual policy cutoff (placebo robustness re-runs the analysis at `placebo_cutoffs`).

placebo_cutoffs	Numeric vector of false cutoffs to test.
bandwidth	Numeric; the local-polynomial bandwidth on each side of the cutoff. NULL invokes the data-driven CCT selector.
p	Integer; local-polynomial order (default 1 for local- linear). 2 picks up quadratic curvature for bias correction.
kernel	One of "triangular" (default), "epanechnikov", "uniform", or "gaussian".
alpha	Significance level (default 0.05).

---

morie_rdd_plot_data *Binned scatter + global-polynomial data for an RD plot*

Description

Binned scatter + global-polynomial data for an RD plot

Usage

```
morie_rdd_plot_data(
  data,
  outcome,
  running,
  cutoff = 0,
  n_bins = 20,
  p_global = 4,
  p_local = 1,
  bandwidth = NULL,
  kernel = "triangular"
)
```

Arguments

data	A data.frame holding the outcome, running variable, treatment, and any covariates referenced by name.
outcome	Character; column name of the response variable in data.
running	Character; column name of the running (forcing) variable in data.
cutoff	Numeric scalar; the threshold on running. Default 0 (the canonical normalisation).
n_bins	Integer; bin count for histogram-based density tests and binned-plot reductions.
p_global	Integer; polynomial order for the global component of morie_rdd_plot_data.
p_local	Integer; polynomial order for the local component of morie_rdd_plot_data.
bandwidth	Numeric; the local-polynomial bandwidth on each side of the cutoff. NULL invokes the data-driven CCT selector.
kernel	One of "triangular" (default), "epanechnikov", "uniform", or "gaussian".

morie_rdd_power *RDD power calculation*

Description

RDD power calculation

Usage

```
morie_rdd_power(
  n,
  tau,
  sigma,
  cutoff_density = 1,
  bandwidth = NULL,
  kernel = "triangular",
  alpha = 0.05
)
```

Arguments

`n`	Integer; sample-size argument to `morie_rdd_power`.
`tau`	Numeric; the treatment-effect size used by power / sample-size calculators.
`sigma`	Numeric; outcome standard deviation.
`cutoff_density`	Numeric; running-variable density at the cutoff.
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

morie_rdd_power_calc *Power calculations for RDD designs via* **rdpower**

Description

Thin extender over `rdpower::rdpower` for sharp / fuzzy RDD power analysis (Cattaneo, Titiunik & Vazquez-Bare, 2019).

Usage

```
morie_rdd_power_calc(data, cutoff = 0, ...)
```

Arguments

`data`	Numeric matrix or data frame with two columns: the outcome Y and the running variable R (as expected by `rdpower::rdpower`'s `data` argument).
`cutoff`	Numeric scalar; the cutoff for the running variable (default `0`).
`...`	Further arguments forwarded to `rdpower::rdpower` (e.g. `tau`, `nsamples`, `kernel`, `vce`, `alpha`, `rho`).

Details

Named `morie_rdd_power_calc` rather than `morie_rdd_power` because the latter is already taken in `R/rdd.R` by a closed-form analytical power formula that takes scalar `(n, tau, sigma)` rather than a data frame; this wrapper preserves that function and offers the full **rdpower** simulation-based surface alongside it.

Value

A list with `$method = "rdpower::rdpower"` and `$raw` (an `rdpower` object with the simulated power and effective sample sizes).

Examples

```
## Not run:
  if (requireNamespace("rdpower", quietly = TRUE)) {
    set.seed(1)
    R <- runif(500, -1, 1)
    Y <- 0.4 * R + (R >= 0) * 0.2 + rnorm(500, sd = 0.5)
    morie_rdd_power_calc(cbind(Y, R), cutoff = 0, tau = 0.2)
  }

## End(Not run)
```

---

morie_rdd_sample_size

RDD sample-size determination

Description

RDD sample-size determination

Usage

```
morie_rdd_sample_size(
  tau,
  sigma,
  cutoff_density = 1,
  bandwidth = 1,
  power = 0.8,
  kernel = "triangular",
```

```
  alpha = 0.05
)
```

Arguments

`tau`	Numeric; the treatment-effect size used by power / sample-size calculators.
`sigma`	Numeric; outcome standard deviation.
`cutoff_density`	Numeric; running-variable density at the cutoff.
`bandwidth`	Numeric; the local-polynomial bandwidth on each side of the cutoff. `NULL` invokes the data-driven CCT selector.
`power`	Numeric in `(0, 1)`; target statistical power.
`kernel`	One of `"triangular"` (default), `"epanechnikov"`, `"uniform"`, or `"gaussian"`.
`alpha`	Significance level (default `0.05`).

`morie_rdd_sharp` *Sharp RDD treatment effect at the cutoff*

Description

Sharp RDD treatment effect at the cutoff

Usage

```
morie_rdd_sharp(
  data,
  outcome,
  running,
  cutoff = 0,
  bandwidth = NULL,
  p = 1,
  kernel = "triangular",
  cluster = NULL,
  covariates = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Data frame.
`outcome`	Outcome column.
`running`	Running variable column.
`cutoff`	Threshold (default 0).
`bandwidth`	Optional bandwidth; if `NULL`, CCT MSE-optimal.

p	Local-polynomial order.
kernel	Kernel name.
cluster	Optional cluster column.
covariates	Optional character vector of covariate names.
alpha	Significance level.

`morie_read_outputs_manifest` *Read outputs manifest from a project*

Description

Read outputs manifest from a project

Usage

```
morie_read_outputs_manifest(
  project_root = NULL,
  manifest_path = NULL,
  validate = TRUE
)
```

Arguments

project_root	Project root path.
manifest_path	Optional explicit manifest path.
validate	If `TRUE`, validate schema.

Value

Manifest data frame.

Examples

```
# Craft a minimal manifest in tempdir and read it back:
tdir <- tempfile("morie-doc-")
dir.create(tdir)
man <- file.path(tdir, "outputs_manifest.csv")
write.csv(
  data.frame(
    output = "results.csv",
    public_path = file.path(tdir, "results.csv"),
    size_kb = 0.01, modified = format(Sys.Date())
  ),
  man,
  row.names = FALSE
)
writeLines("x,y\n1,2", file.path(tdir, "results.csv"))
morie_read_outputs_manifest(manifest_path = man)
```

`morie_recommended_pair_test`
Recommended bivariate test for a pair of variables.

Description

Looks up the (level_a, level_b) combination and returns the right default test (Stevens-1946 hierarchy).

Usage

```
morie_recommended_pair_test(tax_a, tax_b)
```

Arguments

`tax_a`, `tax_b`	Two `morie_variable_taxonomy` objects.

Value

Character scalar — recommended test name.

`morie_recommended_summary`
Recommended summary statistic for a single variable.

Description

Recommended summary statistic for a single variable.

Usage

```
morie_recommended_summary(tax)
```

Arguments

`tax`	A `morie_variable_taxonomy`.

Value

Character scalar — plain-language hint at which summary suits.

morie_regime_switching

Markov-switching regression (Hamilton 1989)

Description

Fit a constant-mean, switching-variance K-regime Markov-switching model by EM (Hamilton filter).

Usage

```
morie_regime_switching(x, k_regimes = 2)
```

Arguments

`x`	Numeric univariate series.
`k_regimes`	Number of latent regimes. Default 2.

Value

Named list with `mu`, `sigma`, `transition`, `smoothed_probabilities`, `loglik`, `n`, `k_regimes`, `method`.

Examples

```
morie_regime_switching(x = rnorm(50))
```

morie_regularization_path

Ridge / LASSO / ElasticNet regularization path (R parity)

Description

Wraps `glmnet::glmnet`. Returns the coefficient path across the supplied `alphas` (lambda grid in glmnet terminology).

Usage

```
morie_regularization_path(
  x,
  y,
  penalty = c("ridge", "lasso", "elasticnet"),
  alphas = NULL,
  l1_ratio = 0.5
)
```

Arguments

`x`	Numeric matrix of predictors.
`y`	Numeric response.
`penalty`	One of "ridge", "lasso", "elasticnet".
`alphas`	Lambda grid. Defaults to a logspace.
`l1_ratio`	glmnet alpha; only used when penalty = "elasticnet".

Value

Named list: estimate, coef_path, alphas, penalty, l1_ratio, n, method.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_risk_difference_ci`

Risk difference (ARD) with Newcombe CI

Description

Risk difference (ARD) with Newcombe CI

Usage

```
morie_risk_difference_ci(table_2x2, alpha = 0.05)
```

Arguments

`table_2x2`	A 2x2 matrix: rows are exposure, columns are outcome.
`alpha`	Significance level.

Value

Named list: `rd`, `ci_lower`, `ci_upper`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_risk_ratio_ci *Risk ratio (relative risk) with log-normal CI*

Description

Risk ratio (relative risk) with log-normal CI

Usage

```
morie_risk_ratio_ci(table_2x2, alpha = 0.05)
```

Arguments

table_2x2	A 2x2 matrix: rows are exposure, columns are outcome (disease = col 1).
alpha	Significance level.

Value

Named list: `rr`, `ci_lower`, `ci_upper`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_rkhs_full *RKHS regression with Gaussian kernel*

Description

RKHS regression with Gaussian kernel

Usage

```
morie_rkhs_full(x, y, markers, h = NULL, lam = 1)
```

Arguments

x	Fixed-effect design.
y	Numeric response.
markers	Genotype matrix (n x m).
h	Kernel bandwidth (default median ||m_i - m_j||^2).
lam	Ridge regulariser on alpha (default 1).

Value

list(estimate, alpha, beta, K, f_hat, se, h, n, method).

References

Gianola & van Kaam (2008). Montesinos Lopez Ch 5.

Examples

```
morie_rkhs_full(x = rnorm(50), y = rnorm(50), markers = matrix(sample(0:2, 200, TRUE), 50, 4))
```

---

morie_rnn_genomic *Vanilla RNN genomic predictor (BPTT, base R)*

Description

Vanilla RNN genomic predictor (BPTT, base R)

Usage

```
morie_rnn_genomic(
  x,
  y,
  markers,
  hidden = 8,
  n_epochs = 150,
  lr = 0.01,
  l2 = 0.001,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	Optional fixed-effect design.
`y`	Numeric response.
`markers`	(n x L) marker sequence.
`hidden`, `n_epochs`, `lr`, `l2`, `seed`	Hyperparameters.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("rnnge", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, y_hat, W_h, W_x, b_h, w_o, b_o, se, n, method).

References

Montesinos Lopez Ch 14.

Examples

```
morie_rnn_genomic(x = rnorm(50), y = rnorm(50), markers = matrix(sample(0:2, 200, TRUE), 50, 4))
```

---

morie_roc_auc_score *ROC curve and AUC (R parity)*

Description

Wraps `pROC::roc`.

Usage

```
morie_roc_auc_score(y_true, y_score)
```

Arguments

`y_true`	Binary labels.
`y_score`	Predicted scores for the positive class.

Value

Named list: estimate, auc, fpr, tpr, thresholds, n, n_positive, n_negative, method.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

---

morie_rsample_bootstraps *Direct bridge to* `rsample::bootstraps`

Description

Thin pass-through that builds a tidymodels-style `rset` of bootstrap resamples via `rsample::bootstraps` and returns it untouched.

Usage

```
morie_rsample_bootstraps(data, times = 25L, ...)
```

Arguments

`data`	A data.frame.
`times`	Number of bootstrap resamples (default 25).
`...`	Forwarded to `rsample::bootstraps` (e.g. `strata`, `apparent`).

Value

An `rset` **rsample** object.

See Also

`rsample::bootstraps`, `rsample::vfold_cv`.

`morie_run_ebac_selection_ipw_analysis` *Run the eBAC selection-adjusted IPW workflow*

Description

Mirrors the core outputs of the old `07_ebac_ipw.R` workflow.

Usage

```
morie_run_ebac_selection_ipw_analysis(
  data,
  output_dir = NULL,
  treatment = "cannabis_any_use",
  covariates = c("age_group", "gender", "province_region", "mental_health",
    "physical_health")
)
```

Arguments

`data`	Analysis data frame.
`output_dir`	Optional directory for CSV outputs.
`treatment`	Treatment column name.
`covariates`	Covariate names used in the observation model.

Value

Named list of output tables and the observed-domain analysis frame.

Examples

```
# Run on a synthetic CPADS-shaped frame (the CKAN-fetched PUMF works
# identically -- see morie_load_cpads_data() for the real frame):
if (requireNamespace("survey", quietly = TRUE)) {
  set.seed(1)
  n <- 200
  cpads <- data.frame(
    weight = runif(n, 0.5, 2),
    alcohol_past12m = rbinom(n, 1, 0.8),
    heavy_drinking_30d = rbinom(n, 1, 0.3),
    ebac_tot = abs(rnorm(n, 0.05, 0.03)),
    ebac_legal = rbinom(n, 1, 0.7),
    cannabis_any_use = rbinom(n, 1, 0.3),
    age_group = sample(1:6, n, TRUE),
    gender = sample(1:2, n, TRUE),
    province_region = sample(1:5, n, TRUE),
    mental_health = sample(1:5, n, TRUE),
    physical_health = sample(1:5, n, TRUE)
  )
  morie_run_ebac_selection_ipw_analysis(cpads)
}
```

---

morie_run_morie_module

Run one implemented MORIE module against CPADS data

Description

Run one implemented MORIE module against CPADS data

Usage

```
morie_run_morie_module(
  module_name,
  cpads_csv = .cpads_default_csv(),
  output_dir = NULL
)
```

Arguments

`module_name`	Module name.
`cpads_csv`	Path to the CPADS CSV.
`output_dir`	Optional directory for CSV outputs.

Value

Named list of data-frame outputs.

Examples

```
# Dispatch one MORIE module against the canonical CPADS CSV. The CSV
# ships with a morie project tree, or is fetched via the CKAN endpoint
# (morie_load_dataset("ocp21")). Wrapped in tryCatch so the example
# documents usage even when the CSV is not checked out locally.
tryCatch(
  morie_run_morie_module("descriptive-statistics"),
  error = function(e) message(conditionMessage(e))
)
```

---

morie_run_morie_modules

Run multiple implemented MORIE modules

Description

Run multiple implemented MORIE modules

Usage

```
morie_run_morie_modules(
  modules = morie_list_morie_modules()$name,
  cpads_csv = .cpads_default_csv(),
  output_dir = NULL
)
```

Arguments

`modules`	Character vector of module names.
`cpads_csv`	Path to the CPADS CSV.
`output_dir`	Optional directory for CSV outputs.

Value

Named list of module outputs.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_run_pipeline` *Run multiple workflow steps*

Description

Run multiple workflow steps

Usage

```
morie_run_pipeline(
  steps = NULL,
  project_root = NULL,
  script_map = morie_default_workflow_map(),
  stop_on_error = TRUE,
  verbose = TRUE
)
```

Arguments

`steps`	Ordered vector of workflow step names.
`project_root`	Project root directory.
`script_map`	Named character vector mapping steps to script paths.
`stop_on_error`	If `TRUE`, stop at first failure.
`verbose`	If `TRUE`, streams command output.

Value

Data frame of step statuses.

Examples

```
# Build a one-step pipeline in tempdir and dispatch it. The
# real package's morie_default_workflow_map() points at scripts that
# live in a morie project tree.
tdir <- tempfile("morie-doc-")
dir.create(tdir)
step <- file.path(tdir, "step.R")
writeLines('cat("hello from pipeline\\n")', step)
morie_run_pipeline(
  steps = "demo",
  project_root = tdir,
  script_map = c(demo = step),
  verbose = FALSE
)
```

`morie_run_propensity_ipw_analysis` *Run the CPADS propensity/IPW workflow*

Description

Mirrors the core outputs of the old `07_propensity.R` workflow.

Usage

```
morie_run_propensity_ipw_analysis(
  data,
  output_dir = NULL,
  trim = c(0.01, 0.99),
  treatment = "cannabis_any_use",
  outcome = "heavy_drinking_30d",
  covariates = c("age_group", "gender", "province_region", "mental_health",
    "physical_health")
)
```

Arguments

`data`	Analysis data frame.
`output_dir`	Optional directory for CSV outputs.
`trim`	Quantile pair used to trim extreme IPW values.
`treatment`	Binary treatment column.
`outcome`	Binary outcome column.
`covariates`	Covariate names for the propensity model.

Value

Named list of output tables and the analysis data.

Examples

```
# Run on a synthetic CPADS-shaped frame (the CKAN-fetched PUMF works
# identically -- see morie_load_cpads_data() for the real frame):
set.seed(1)
n <- 200
cpads <- data.frame(
  weight = runif(n, 0.5, 2),
  alcohol_past12m = rbinom(n, 1, 0.8),
  heavy_drinking_30d = rbinom(n, 1, 0.3),
  ebac_tot = abs(rnorm(n, 0.05, 0.03)),
  ebac_legal = rbinom(n, 1, 0.7),
  cannabis_any_use = rbinom(n, 1, 0.3),
  age_group = sample(1:6, n, TRUE),
```

```
  gender = sample(1:2, n, TRUE),
  province_region = sample(1:5, n, TRUE),
  mental_health = sample(1:5, n, TRUE),
  physical_health = sample(1:5, n, TRUE)
)
morie_run_propensity_ipw_analysis(cpads)
```

`morie_run_treatment_effects_analysis` *Run a treatment-effects analysis (point estimate, SE, 95% CI)*

Description

Mirrors the Python `morie.run_treatment_effects_analysis()`. Convenience wrapper around `morie_estimate_ate()` that also produces a 95% confidence interval (delta-method approximation).

R port of `investigation.run_treatment_effects_analysis`. Returns a list with:

- `treatment_effects_summary` – data.frame of ATE / ATT / ATC point estimates (SE / CI columns present but NA, matching Python).
- `cate_subgroup_estimates` – data.frame of within-subgroup Hajek IPW CATEs with sandwich-style SEs and Wald 95\% CIs.
- `analysis_frame` – the trimmed data.frame with attached propensity score and IPW weight columns (`ps`, `w_ate`, `w_att`, `w_atc`).
- Convenience scalars `ate` / `att` / `atc` / `se` / `ci_lower` / `ci_upper` / `n` / `method` preserved from the previous R surface for backward compatibility.

Usage

```
morie_run_treatment_effects_analysis(data, treatment, outcome, covariates)

morie_run_treatment_effects_analysis(data, treatment, outcome, covariates)
```

Arguments

`data`	data.frame containing treatment, outcome, and covariates.
`treatment`	Treatment column name.
`outcome`	Outcome column name.
`covariates`	Character vector of covariate column names.

Value

A list with `ate`, `se`, `ci_lower`, `ci_upper`, `n`, `method`.

Named list as described above.

Examples

```
set.seed(1)
df <- data.frame(
  y = rnorm(200),
  t = rbinom(200, 1, 0.5),
  x1 = rnorm(200), x2 = rnorm(200)
)
morie_run_treatment_effects_analysis(df,
  treatment = "t", outcome = "y", covariates = c("x1", "x2")
)
```

---

morie_run_weighted_logistic_analysis

Run a weighted logistic-regression analysis

Description

Mirrors the Python `morie.run_weighted_logistic_analysis()`. Fits a binary-outcome model using survey weights via `survey::svyglm()` if the suggested `survey` package is available, otherwise falls back to base `glm()` with case weights.

Usage

```
morie_run_weighted_logistic_analysis(
  data,
  outcome,
  predictors,
  weights_col = NULL
)
```

Arguments

`data`	A `data.frame` containing outcome, predictors, and (optionally) a weights column.
`outcome`	Column name of the binary outcome.
`predictors`	Character vector of predictor column names.
`weights_col`	Optional column name of analytical weights.

Value

A list with components `coefficients` (named numeric vector), `std_errors`, `p_values`, `n`, `method` ("svyglm" or "glm-weighted").

Examples

```
set.seed(1)
df <- data.frame(
  y = rbinom(200, 1, 0.4),
  x1 = rnorm(200),
  x2 = rnorm(200),
  w = runif(200, 0.5, 1.5)
)
morie_run_weighted_logistic_analysis(df,
  outcome = "y", predictors = c("x1", "x2"), weights_col = "w"
)
```

---

morie_run_workflow_step

Run one project workflow step

Description

Run one project workflow step

Usage

```
morie_run_workflow_step(
  step,
  project_root = NULL,
  script_map = morie_default_workflow_map(),
  rscript_bin = file.path(R.home("bin"), "Rscript"),
  verbose = TRUE
)
```

Arguments

`step`	Step name present in `script_map`.
`project_root`	Project root directory.
`script_map`	Named character vector mapping steps to script paths.
`rscript_bin`	Optional path to `Rscript` binary.
`verbose`	If `TRUE`, streams command output.

Value

Named list with step metadata and exit status.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_sample *Load a bundled MORIE reference sample by name*

Description

Returns a small CSV that ships with the package, suitable for running examples and tests of the `mrm_*()` callables without any network or external data dependency.

Usage

```
morie_sample(name = c("otis_b01", "otis_b09", "otis_c11", "tps_assault"))
```

Arguments

`name`	One of `"otis_b01"`, `"otis_b09"`, `"otis_c11"`, `"tps_assault"`.

Value

A data.frame.

Examples

```
b01 <- morie_sample("otis_b01")
head(b01)
```

morie_sample_size_logistic *Sample size for logistic regression detecting a target odds ratio*

Description

Uses the formula from Hsieh et al. (1998):

$$n = \frac{(z_{\alpha/2} + z_{\beta})^2}{p_1(1 - p_1)[\log(OR)]^2}$$

Usage

```
morie_sample_size_logistic(p0, or, alpha = 0.05, power = 0.8, two_sided = TRUE)
```

Arguments

`p0`	Prevalence under control.
`or`	Target odds ratio.
`alpha`	Significance level.
`power`	Desired power.
`two_sided`	Logical.

Value

Integer sample size.

References

Hsieh FY, Bloch DA, Larsen MD (1998). A simple method of sample size calculation for linear and logistic regression. *Statistics in Medicine*, 17(14):1623-1634.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_semipar_bridge` *Semiparametric kernel primitives (R port)*

Description

R port of `morie.semipar_bridge`. Provides the kernel-based building blocks used by morie's nuisance estimation pipelines (TMLE, AIPW, DML): kernel evaluation, Nadaraya-Watson regression, local linear regression, kernel density estimation, and bandwidth selection.

Details

The Python module loads a C shared library (`semipar_kernels.dylib` / `.so`) and falls back to NumPy. The R port is pure R: it implements the same algorithms in vectorised form and additionally wraps `mgcv::gam` for a high-quality penalised-spline smoother as an alternative to manual bandwidth selection.

Functions:

- `kernel_eval`: evaluate a kernel function.
- `nw_regression`: Nadaraya-Watson kernel regression.
- `local_linear`: local linear kernel regression.
- `kde`: kernel density estimation.
- `silverman_bandwidth`: rule-of-thumb bandwidth.
- `loocv_bandwidth`: leave-one-out CV bandwidth for NW regression.
- `kernel_cond_moments`: kernel-weighted mean and variance.
- `gam_smoother`: `mgcv::gam` thin-plate smoother fit + predict.
- `SemiparKernels`: object-style wrapper.

`morie_sensitivity_evalue` *E-values for the EValue dispatch family (extender)*

Description

Thin interface to the **EValue** dispatch family (`evalues.OLS`, `evalues.RR`, `evalues.OR`, `evalues.HR`, `evalues.MD`), exposed under the `morie_sensitivity_*` namespace so MRM / paper callers can reach the full **EValue** surface without loading **EValue** directly. Pairs with `e_value_rr` / `e_value_or` / `e_value_hr` / `e_value_d`, which are the typed convenience wrappers around the same backend.

Usage

```
morie_sensitivity_evalue(
  estimate,
  se = NULL,
  sd = NULL,
  type = c("OLS", "RR", "OR", "HR", "MD"),
  rare = TRUE,
  true = NULL,
  ci_lower = NULL,
  ci_upper = NULL,
  ...
)
```

Arguments

`estimate`	Observed effect on the requested scale.
`se`	Standard error of `estimate` (used by `"OLS"` and `"MD"`).
`sd`	Outcome standard deviation (required by `"OLS"`).
`type`	One of `"OLS"` (default), `"RR"`, `"OR"`, `"HR"`, `"MD"`.
`rare`	Logical; only relevant for `"OR"` and `"HR"` (passed to **EValue**). Default `TRUE` (rare-outcome approximation).
`true`	Reference value on the appropriate scale (default 0 for OLS / MD, 1 for ratio scales).
`ci_lower`, `ci_upper`	Optional 95% CI on the same scale as `estimate`.
`...`	Additional arguments forwarded to the underlying `EValue::evalues.*` function.

Value

A list of class `morie_sensitivity_evalue` with `estimate`, `e_value_point`, `e_value_ci`, `type`, `method`, and `raw` (the full EValue matrix).

References

VanderWeele, T. J., & Ding, P. (2017). Sensitivity analysis in observational research: introducing the E-value. *Annals of Internal Medicine*, 167(4), 268–274.

`morie_sensitivity_konfound`
Konfound robustness for a coefficient (konfound extender)

Description

Thin interface to `konfound::pkonfound`: how many cases would need to be replaced with average-treatment-effect cases (or how large would an omitted-variable correlation have to be) to invalidate the inference? Pairs with `morie_sensitivity_omitted_var_bias` (which uses the Cinelli-Hazlett partial-R-squared framing instead of the Frank et al. percent-bias-to-invalidate framing).

Usage

```
morie_sensitivity_konfound(
  estimate,
  se,
  n,
  n_covariates = 0L,
  alpha = 0.05,
  ...
)
```

Arguments

`estimate`	Treatment-coefficient estimate.
`se`	Standard error of `estimate`.
`n`	Number of observations.
`n_covariates`	Number of covariates in the model (excluding the intercept and the treatment). Default 0.
`alpha`	Significance level. Default 0.05.
`...`	Additional arguments forwarded to `konfound::pkonfound`.

Value

A list of class `morie_sensitivity_konfound` with the percent-bias-to-invalidate, the impact-threshold-of-a- confounding-variable (ITCV), and the raw konfound object.

References

Frank, K. A., Maroulis, S. J., Duong, M. Q., & Kelcey, B. M. (2013). What would it take to change an inference? *Educational Evaluation and Policy Analysis*, 35(4), 437–460.

`morie_sensitivity_omitted_var_bias` *Omitted-variable bias on a fitted model (sensemakr extender)*

Description

Thin interface to `sensemakr::sensemakr`: returns the full Cinelli-Hazlett robustness-value object including benchmark bounds, adjusted t-statistics, and the data needed to draw contour plots. Pairs with `omitted_variable_bias`, which is the closed-form version that takes `estimate` + `se` + degrees of freedom directly (useful when you don't have an `lm` object handy).

Usage

```
morie_sensitivity_omitted_var_bias(
  model,
  treatment,
  benchmark_covariates = NULL,
  kd = c(1, 2, 3),
  ky = NULL,
  q = 1,
  alpha = 0.05,
  ...
)
```

Arguments

`model`	A fitted regression model (`lm` or compatible).
`treatment`	Name of the treatment variable (coefficient).
`benchmark_covariates`	Optional character vector of covariate names whose strengths bound the unmeasured-confounder strength.
`kd`	Multipliers on the benchmark covariate strength. Default `c(1, 2, 3)`.
`ky`	Multipliers on the benchmark covariate's outcome strength. Default equal to `kd`.
`q`	Fraction of the estimate to be explained away. Default 1.
`alpha`	Significance level. Default 0.05.
`...`	Additional arguments forwarded to `sensemakr::sensemakr`.

Value

A list of class `morie_sensitivity_omitted_var_bias` with the robustness values, partial R-squared of treatment, benchmark bounds, and the full sensemakr object as `raw`.

References

Cinelli, C., & Hazlett, C. (2020). Making sense of sensitivity: extending omitted variable bias. *Journal of the Royal Statistical Society B*, 82(1), 39–67.

`morie_sensitivity_rosenbaum`
Rosenbaum bounds sensitivity analysis

Description

For a range of hidden-confounding levels Γ, tests whether the treatment effect remains significant. A large Γ at which the result remains significant indicates robustness.

Usage

```
morie_sensitivity_rosenbaum(
  treated,
  control,
  gamma_range = seq(1, 3, by = 0.2)
)
```

Arguments

`treated`	Numeric vector of outcomes for treated units.
`control`	Numeric vector of outcomes for control units (may differ in length from `treated` for unmatched designs).
`gamma_range`	Numeric vector of Γ values to test.

Details

Delegates to `rbounds::psens()` when **rbounds** is installed and pairs-of-equal-length data are supplied; alternatively delegates to `sensitivitymv::senmv()` when **sensitivitymv** is installed. Otherwise falls back to inline sign-score bounds (Rosenbaum 2002, Section 4.3).

Value

Data frame with columns: `gamma`, `p_lower`, `p_upper`.

References

Rosenbaum PR (2002). *Observational Studies* (2nd ed.). Springer.

Examples

```
morie_sensitivity_rosenbaum(treated = rnorm(30, 0.5), control = rnorm(30))
```

morie_sensitivity_tipping_point *Tipping-point sensitivity to a single unmeasured confounder (tipr)*

Description

Thin interface to `tipr::tip`: returns the minimum value of the standardised mean difference (`smd`) or partial R-squared (`R2`) of an unmeasured confounder that would tip the lower (or upper) bound of the confidence interval back to the null. Pairs with `tipping_point_analysis`, which targets *missing-data* sensitivity rather than unmeasured-confounder sensitivity.

Usage

```
morie_sensitivity_tipping_point(estimate, smd = NULL, r2 = NULL, ...)
```

Arguments

`estimate`	Observed treatment effect on the coefficient scale.
`smd`	Hypothesised standardised mean difference of the unmeasured confounder between treatment groups.
`r2`	Hypothesised partial R-squared of the unmeasured confounder with the outcome. Forwarded as the `r_squared` tipr argument.
`...`	Additional arguments forwarded to `tipr::tip` (e.g. `outcome_type`, `confidence`).

Value

A list of class `morie_sensitivity_tipping_point` with the tipped point estimate and the raw **tipr** object.

References

D'Agostino McGowan, L. (2022). tipr: An R package for sensitivity analyses for unmeasured confounders. *Journal of Open Source Software*, 7(77), 4495.

morie_sgolay_smooth *Savitzky-Golay smoothing filter*

Description

Polynomial-fit smoothing filter. Preserves higher moments (peak heights, shape) better than a moving average and is the standard tool for chromatography, spectroscopy, and biosensor smoothing.

Usage

```
morie_sgolay_smooth(x, window_length = 11L, polyorder = 3L)
```

Arguments

`x`	Numeric vector.
`window_length`	Window length (odd integer, default 11).
`polyorder`	Polynomial order (default 3).

Value

List with `filtered` (numeric vector) and `name`.

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  t <- seq(0, 1, length.out = 200)
  x <- sin(2 * pi * 3 * t) + rnorm(200, sd = 0.2)
  y <- morie_sgolay_smooth(x, window_length = 11, polyorder = 3)
  length(y$filtered)
}
```

`morie_shapiro_wilk_test`

Shapiro-Wilk normality test

Description

Shapiro-Wilk normality test

Usage

```
morie_shapiro_wilk_test(x, alpha = 0.05)
```

Arguments

`x`	Numeric vector.
`alpha`	Significance level for the `is_normal` flag (default 0.05).

Value

Named list: `W`, `p_value`, `is_normal`.

Examples

```
morie_shapiro_wilk_test(x = rnorm(50))
```

`morie_sign_test_power`
Power function of the two-sided sign test (Gibbons Ch 5.4.4)

Description

Builds the discrete rejection region under H0: p = 0.5 with size <= alpha, then evaluates power at the alternative p_alt.

Usage

```
morie_sign_test_power(x, mu0 = 0, p_alt = 0.7, alpha = 0.05)
```

Arguments

`x`	Numeric vector (only `length(x != mu0)` is used).
`mu0`	Null median.
`p_alt`	Alternative success probability P(X > mu0).
`alpha`	Test level.

Value

Named list: statistic (power), n, p_alt, alpha, size, k_lower, k_upper.

Examples

```
morie_sign_test_power(x = rnorm(50))
```

`morie_simple_random_sample`
Simple random sample from a data frame

Description

Simple random sample from a data frame

Usage

```
morie_simple_random_sample(df, n, replace = FALSE, seed = 42L)
```

Arguments

`df`	A data frame.
`n`	Number of units to select.
`replace`	Sample with replacement? Default `FALSE`.
`seed`	Random seed for reproducibility.

Value

A data frame of `n` sampled rows with a `.weight` column added.

Examples

```
df <- data.frame(x = 1:100)
srs_sample <- morie_simple_random_sample(df, 20)
```

`morie_simpleboot_two` *Direct bridge to* `simpleboot::two.boot`

Description

Thin pass-through to `simpleboot::two.boot` for the common two-sample bootstrap-of-a-statistic case (e.g. difference of means, ratio of medians). Returns a `boot` object the user can hand to `boot::boot.ci` or `morie_boot_basic_ci()`.

Usage

```
morie_simpleboot_two(x, y, statistic = mean, R = 1000L, ...)
```

Arguments

`x`, `y`	Numeric vectors.
`statistic`	A scalar function applied to one sample at a time (e.g. `mean`, `median`); two.boot's contract.
`R`	Number of bootstrap replicates (default 1000).
`...`	Forwarded to `simpleboot::two.boot`.

Value

A `boot` object.

See Also

`simpleboot::two.boot`, `simpleboot::one.boot`, `simpleboot::lm.boot`.

`morie_simulate_longitudinal_panel` *Simulate a longitudinal panel and return a tidy long-format data.frame*

Description

Simulate a longitudinal panel and return a tidy long-format data.frame

Usage

```
morie_simulate_longitudinal_panel(
  n_individuals = 50,
  n_timepoints = 20,
  p_variables = 3,
  cov_kernel = "ar1",
  cov_rho = 0.5,
  ar_lags = 1L,
  ar_spectral_radius = 0.8,
  ar_decay = 0.6,
  missing_fraction = 0,
  outlier_fraction = 0,
  outlier_scale = 5,
  seed = 42L
)
```

Arguments

`n_individuals`	Number of subjects.
`n_timepoints`	Number of time-points per subject.
`p_variables`	Number of variables.
`cov_kernel`	Innovation covariance kernel.
`cov_rho`	Correlation parameter.
`ar_lags`	VAR lag order.
`ar_spectral_radius`	Target spectral radius (per lag).
`ar_decay`	Geometric decay across lags.
`missing_fraction`	Probability of NA mask per entry.
`outlier_fraction`	Probability of outlier amplification.
`outlier_scale`	Multiplicative factor for outliers.
`seed`	Non-negative integer seed.

Value

A data.frame with columns subject_id, t, variable, value.

Examples

```
if (FALSE) {
  df <- morie_simulate_longitudinal_panel(
    n_individuals = 30, n_timepoints = 10, p_variables = 4
  )
  head(df)
}
```

---

morie_siu_all_analyses

Run every SIU descriptive analysis in this module

Description

Convenience wrapper that runs each of the analysis surfaces and optionally writes a `.txt` dump (and `.json` payload if **jsonlite** is available) per result. Failures are captured per-analysis: one bad surface does not stop the rest.

Usage

```
morie_siu_all_analyses(data = NULL, out_dir = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.
`out_dir`	Optional directory; if non-`NULL`, each result is written as `siu_analysis_<name>.txt`.

Value

A named list of `morie_rich_result` objects.

`morie_siu_analyze` *SIU descriptive analyses (R port of* `morie.siu.analyze`*)*

Description

Turns a scraped SIU_by_case.csv (the canonical output of morie_fetch_siu) into a set of RichResult-style descriptive analyses. Each callable accepts either a path to the CSV or a data frame directly and returns a structured list with `title`, `summary_lines`, `tables`, `interpretation`, `warnings`, and `payload` fields. The structure mirrors the Python analyze.py surfaces so that the same multi-section render works through morie's rich-output dispatch.

Details

These analyses are deliberately distinct from the published-table replicators in morie_siu_sprott_doob_feb2021 and the Mandela classifier in morie_siu_classify_mandela: those work from CSC SIU person-stay data, whereas these analyses work from the Ontario SIU *director's-report* corpus.

See Also

morie_fetch_siu, morie_siu_classify_mandela, mrm_siu_per_service_rate.

`morie_siu_anomaly_check`
Per-field anomaly check: does the parser's extraction match the HTML?

Description

For each populated field in the parser's row, ask the LLM whether the extracted value is supported by the cached report HTML. Used to surface fields where the regex parser is plausibly wrong – the LLM's verdicts are not authoritative, just an automated way to triage which rows a human should re-read against the HTML.

Usage

```
morie_siu_anomaly_check(
  case_number,
  model = c("ollama", "gemini"),
  cache_dir = file.path(tempdir(), "morie", "siu"),
  max_html_chars = 80000L,
  mock_response_text = NULL
)
```

Arguments

`case_number`	An SIU case number (e.g. `"17-OVI-201"`).
`model`	One of `"ollama"` (default; free, runs locally, zero-config when an Ollama daemon is on `localhost:11434`), `"gemini"` (paid), or `"claude"` (paid). A character vector enables fail-over: the first model whose call succeeds wins. The default `c("ollama", "gemini")` tries the local free model first and only escalates to paid Gemini if Ollama isn't installed or fails – so morie costs $0 to use as long as you have a free Gemma / Qwen / Llama running locally (e.g. `ollama pull gemma3:4b`).
`cache_dir`	Directory holding the harvester's SIU.csv and the optional `html/` subdirectory.
`max_html_chars`	Soft cap on the HTML payload sent to the model (default 80,000 – larger than any real SIU report, small enough to stay under typical context budgets).
`mock_response_text`	For testing only: if non-NULL, skip the network call and use this string as the model's raw reply.

Details

One API call is made per case (all fields batched into a single prompt with structured-JSON output).

Value

A data frame with one row per populated parser field: `field`, `parser_value`, `verdict` (one of `"agree"` / `"disagree"` / `"unclear"`), and `reason` (a short sentence pointing to the report passage).

Examples

```
## Not run:
Sys.setenv(GOOGLE_API_KEY = "your-gemini-key")
a <- morie_siu_anomaly_check("17-OVI-201", model = "gemini")
subset(a, verdict == "disagree")

## End(Not run)
```

`morie_siu_audit_case` *Audit one SIU case end-to-end: parser output + raw HTML*

Description

For any case_number (or drid), return the parser's 64-column row together with the raw HTML pages it was extracted from – the director's-report page and, when linked, the news-release page. This is the per-row ground truth: every field in the emitted CSV is reproducible from `report_html` via the parser, and any disagreement with another data source can be adjudicated against the saved HTML.

Usage

```
morie_siu_audit_case(
  case_number,
  cache_dir = file.path(tempdir(), "morie", "siu"),
  fetch_if_missing = TRUE
)
```

Arguments

case_number	An SIU case number (e.g. "17-OVI-201"), or an integer drid.
cache_dir	Directory holding the harvester's SIU.csv and the optional html/ subdirectory.
fetch_if_missing	If TRUE (default), fetch the page from SIU when the local cache misses. Set FALSE to work strictly from the cache.

Details

Reads from the local cache at <cache_dir>/html/ (populated by morie_fetch_siu(cache_html = TRUE)) when available, and falls back to a polite live fetch when the cache is missing.

Value

A list with elements row (the parser's 1-row data frame for this case), drid, nrid, report_html, news_html, report_text (HTML-stripped plain text of the report) and news_text.

Examples

```
## Not run:
a <- morie_siu_audit_case(
  "17-OVI-201",
  cache_dir = file.path(tempdir(), "morie", "siu")
)
cat(substr(a$report_text, 1, 1000), "\n")

## End(Not run)
```

---

morie_siu_audit_columns

Per-column accuracy audit: estimate every SIU column's correctness

---

Description

Runs morie_siu_anomaly_check() on a vector of case_numbers and aggregates per-field across them. Output is a data frame with one row per SIU column, ordered by how often the LLM auditor agreed with the C++ parser. The worst-ranked rows are the parser fields that most deserve regex / extraction-logic fixes.

Usage

```
morie_siu_audit_columns(
  case_numbers,
  model = c("ollama", "gemini"),
  cache_dir = file.path(tempdir(), "morie", "siu"),
  max_html_chars = 80000L,
  max_examples_per_field = 5L,
  progress = TRUE
)
```

Arguments

`case_numbers`	Character vector of SIU case numbers to audit.
`model`	One of `"ollama"` (default; free, runs locally, zero-config when an Ollama daemon is on `localhost:11434`), `"gemini"` (paid), or `"claude"` (paid). A character vector enables fail-over: the first model whose call succeeds wins. The default `c("ollama", "gemini")` tries the local free model first and only escalates to paid Gemini if Ollama isn't installed or fails – so morie costs $0 to use as long as you have a free Gemma / Qwen / Llama running locally (e.g. `ollama pull gemma3:4b`).
`cache_dir`	Directory holding the harvester's SIU.csv and the optional `html/` subdirectory.
`max_html_chars`	Soft cap on the HTML payload sent to the model (default 80,000 – larger than any real SIU report, small enough to stay under typical context budgets).
`max_examples_per_field`	Maximum disagreement examples retained per field (default 5).
`progress`	Logical; print a per-case progress line.

Details

Examples of LLM-flagged disagreements are attached as the `"examples"` attribute of the returned data frame (one nested data frame per field), with at most `max_examples_per_field` cases each. Each example carries the case_number, the parser_value, and the LLM's one-sentence reason – enough for a maintainer to pop the cached HTML for that case, see who's right, and decide whether to refine the regex pattern for that field.

Designed for cheap local audit: with `model = "ollama"` pointed at a local Gemma / Qwen / DeepSeek instance, auditing 50-100 cases costs zero API spend and finishes in a few minutes. With `model = c("gemini", "ollama")` the chain uses paid Gemini first and silently falls back to the local model on quota / network errors.

Value

A data frame with columns `field`, `n_audited`, `n_agree`, `n_disagree`, `n_unclear`, `agree_rate`. Sorted ascending by `agree_rate` so the most-broken fields land at the top. The `"examples"` attribute holds nested data frames of flagged cases per field.

Examples

```
## Not run:
Sys.setenv(
  OLLAMA_HOST = "http://localhost:11434",
  OLLAMA_MODEL = "gemma3:4b"
)
csv <- morie_fetch_siu(cache_html = TRUE)
df <- utils::read.csv(csv, colClasses = "character")
sample <- sample(df$case_number[nzchar(df$case_number)], 50L)
audit <- morie_siu_audit_columns(sample, model = "ollama")
# Worst 8 fields, ripe for parser fixes:
head(audit, 8)
# See concrete disagreements for the worst field:
attr(audit, "examples")[[audit$field[1L]]]

## End(Not run)
```

---

morie_siu_by_police_service

SIU cases by police service

Description

Per-police-service tabulation of case counts, charges-recommended counts, and the implied charge rate. Sorted by case count (descending), capped at the top 30 in the rendered table.

Usage

```
morie_siu_by_police_service(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list with summary lines, one table, and a payload of raw counts.

morie_siu_by_year *SIU cases by year*

Description

Year-over-year case volume plus charges-rate, parsed from the first four characters of `date_of_incident_iso`.

Usage

```
morie_siu_by_year(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list.

morie_siu_cache_path *Cache-path helper for the lightweight SIU scraper*

Description

Returns the path `<cache_dir>/SIU.csv`, creating `cache_dir` if needed. Default is `file.path(tempdir(),"morie" "siu")`; pass `morie_cache_dir("siu")` for persistent caching.

Usage

```
morie_siu_cache_path(cache_dir = file.path(tempdir(), "morie", "siu"))
```

Arguments

`cache_dir`	Output directory.

Value

Absolute path to `SIU.csv` (file may not exist yet).

Examples

```
p <- morie_siu_cache_path(tempfile("siu_demo_"))
p
```

`morie_siu_case_counts`
SIU per-case team-size distribution

Description

Distribution (n, mean, median, min, max) of subject-officials, witness-officials, civilian-witnesses, and officers-involved.

Usage

```
morie_siu_case_counts(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list.

`morie_siu_charges_by_year_chi2`
Chi-square: charges-recommended independent of year?

Description

Cross-tabulates `charges_recommended` against incident-year and tests independence with a Pearson chi-square (`stats::chisq.test`; p-value via `stats::pchisq`). Years with zero charge-decided cases are dropped. Complements `morie_siu_verify_chi2` in `sprott_doob.R`, which tests specific published 2x2 tables; this is a generic "did the charge rate move?" probe over the harvested SIU corpus.

Usage

```
morie_siu_charges_by_year_chi2(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list including the contingency table, statistic, df, and p-value.

`morie_siu_classify_mandela` *Apply the Sprott-Doob Mandela-Rules classifier to one SIU stay*

Description

Operationalizes UN Mandela Rules 43 and 44 against a single SIU person-stay's days, average hours out of cell, and percent- of-days that missed the legislatively-required 4 hours out of cell.

Usage

```
morie_siu_classify_mandela(
  days_in_siu,
  hrs_out_of_cell_avg,
  missed_full_4hrs_pct_of_days
)
```

Arguments

`days_in_siu`	Length of the stay (days).
`hrs_out_of_cell_avg`	Average hours out of cell per day during the stay.
`missed_full_4hrs_pct_of_days`	Percent of days (0-100) where the inmate did not receive the legislatively-required 4 hours out of cell.

Details

- **Solitary Confinement** (Rule 44): <=2 hrs avg out of cell, missed full 4 hrs every day, stay <=15 days.
- **Torture** (Rules 43+44): same conditions but stay length >=16 days (crosses the "prolonged" threshold).
- **All other**: did not meet either threshold.

Value

A named list with elements `category`, `rule`, and `reason`.

Examples

```
morie_siu_classify_mandela(20, 1.5, 100)$category
morie_siu_classify_mandela(8, 1.5, 100)$category
morie_siu_classify_mandela(20, 5, 50)$category
```

`morie_siu_compare` *Field-by-field SIU comparison against a user-supplied external table*

Description

For one case_number, line up the parser's value against the same field in a user-supplied external data source – and, critically, show the surrounding report HTML so the user can adjudicate any disagreement against the actual source document.

Usage

```
morie_siu_compare(
  case_number,
  external,
  field_map = NULL,
  external_case_col = "Q1",
  cache_dir = file.path(tempdir(), "morie", "siu")
)
```

Arguments

`case_number`	A case number (e.g. `"17-OVI-201"`).
`external`	A data frame of external answers, OR a path to an `.xlsx` file (read with `readxl`). Must contain a column whose values match SIU case numbers (default `external_case_col = "Q1"`).
`field_map`	A named list mapping external-column names to morie field names.
`external_case_col`	Name of the external column carrying the case-number key.
`cache_dir`	Directory holding the harvester's SIU.csv and optional cached HTML.

Details

The ground truth is the SIU director's-report HTML itself. The HTML is what the SIU published; the parser's job is to extract structured fields from it faithfully, and any field's correctness is decidable by reading the cached HTML for that case. Any external reference – a hand-coded survey, an independently-scraped CSV, a colleague's analysis – is just another extraction attempt, possibly with its own errors. This function does not endorse any external source; it only displays both side-by-side with the HTML excerpt so you can decide.

The default field map covers the common SIU-extraction column layout (`Q1 = case_number`, `Q3 = police_service`, `Q4 = number_of_officers_involved`, ...). Pass a custom `field_map` for any other external schema.

Value

A data frame with one row per mapped field: `field`, `parser_value`, `external_value`, `agree`, and `html_excerpt` (a 240-character window around the first occurrence of either value in the cleaned report text). When parser and external disagree, the `html_excerpt` is the tie-breaker.

Examples

```
## Not run:
# Caller supplies their own external table; nothing about the
# mapping or the file format is canonical to morie.
external <- data.frame(case_id = "17-OVI-201", officers = 1L)
cmp <- morie_siu_compare(
  "17-OVI-201",
  external = external,
  field_map = list(officers = "number_of_officers_involved"),
  external_case_col = "case_id"
)
subset(cmp, !agree)

## End(Not run)
```

---

morie_siu_decision_timing

SIU decision-timing distributions

Description

Day-delta distributions for the three SIU process intervals:

Incident -> SIU notified From `date_of_incident_iso` to `date_siu_notified_iso`.

Notified -> director's decision From `date_siu_notified_iso` to `date_of_director_decision_iso`.

Incident -> decision (total) The end-to-end interval.

Usage

```
morie_siu_decision_timing(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list.

`morie_siu_demographics` *Affected-person demographics*

Description

Sex/gender frequency table and age-distribution summary.

Usage

```
morie_siu_demographics(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list.

`morie_siu_fetch` *Lightweight Ontario SIU Director's Reports scraper (R-native)*

Description

On-demand scraper for the Ontario Special Investigations Unit (SIU) Director's Reports index at https://www.siu.on.ca/en/directors_reports.php. This is the R port of `morie.siu_fetch` – the lightweight `httr2`/`rvest` path that complements the C/C++ harvester in `morie_fetch_siu`. Use this when:

Usage

```
morie_siu_index_url()
```

Details

- you want a tiny R-only dependency footprint (no compiled code);
- you only need the header / index fields (case_number, police_service, incident date, decision date) – not the full 64-column schema;
- you are running on a host where the C++ parser does not build.

Distribution policy (2026-05): the scraped corpus is NOT shipped with the package. Each user runs the scraper themselves, which is unambiguously fair use of public oversight reports.

The scraper is conservative: a 2-second delay between requests, retries on 5xx, and a descriptive user-agent. The latest published year as of release is 2023; `years = NULL` (the default) scrapes the unfiltered index, which surfaces the most recent posts.

Cache directory

By default this writes `SIU.csv` under `tempdir()` so R cleans it up at end of session. Pass `cache_dir = morie_cache_dir("siu")` explicitly to opt into a persistent cross- session cache; see `morie_cache_dir` and `morie_cache_clear` (no implicit writes to `~/.cache`).

`morie_siu_fetch_cases` *Scrape Ontario SIU Director's Reports into a tidy CSV*

Description

Pulls the SIU index, walks every linked case-detail page, and writes a six-column CSV (`case_number`, `police_service`, `incident_iso`, `notification_iso`, `decision_iso`, `director_decision_text`, `source_url`) into `cache_dir`.

Usage

```
morie_siu_fetch_cases(
  years = NULL,
  cache_dir = file.path(tempdir(), "morie", "siu"),
  overwrite = FALSE,
  progress = TRUE
)
```

Arguments

`years`	Integer vector of fiscal years to scrape, or `NULL` (default) to scrape the unfiltered index. Years above `2023` (the latest published as of release) may return empty results.
`cache_dir`	Output directory. Default `file.path(tempdir(), "morie", "siu")`; pass `morie_cache_dir("siu")` for persistent caching.
`overwrite`	Logical; if `FALSE` and `SIU.csv` already exists, its path is returned without re-scraping.
`progress`	Logical; print a one-line status per index / case fetch when `TRUE` (default).

Details

This is the lightweight R-only path. For the full 64-column corpus use `morie_fetch_siu` (compiled C++ harvester).

Value

Path to the written `SIU.csv`.

Examples

```
## Not run:
# Network: scrapes the SIU index (~5-15 min at the polite rate).
csv <- morie_siu_fetch_cases(cache_dir = tempfile("siu_"))
utils::head(utils::read.csv(csv))

## End(Not run)
```

---

morie_siu_fetch_dataframe

Scrape Ontario SIU Director's Reports and return a data frame

---

Description

Thin wrapper over morie_siu_fetch_cases, returning a data frame instead of the CSV path. Mirrors the Python `fetch_siu_dataframe()` adapter used by the dataset catalog.

Usage

```
morie_siu_fetch_dataframe(...)
```

Arguments

`...`	Forwarded to `morie_siu_fetch_cases`.

Value

A data frame with the six-column SIU header schema.

Examples

```
## Not run:
df <- morie_siu_fetch_dataframe(cache_dir = tempfile("siu_"))
utils::head(df)

## End(Not run)
```

`morie_siu_index` *SIU drid → case_number → language index*

Description

Returns the shipped drid manifest as a data frame – one row per director's-report id morie has verified, with the parsed case number, detected language, and the canonical drid (the English drid for that case, or the first drid if no English version exists). This is the index `morie_fetch_siu()` uses internally; exposing it lets users:

Usage

```
morie_siu_index(lang = c("all", "en", "fr", "valid"), canonical_only = FALSE)
```

Arguments

`lang`	Filter rows by detected language. `"all"` (default) returns every entry; `"en"` returns only the English drids; `"fr"` returns only French; `"valid"` returns every drid whose case_number was successfully parsed (drops blank / draft drids).
`canonical_only`	If `TRUE`, returns one row per case_number (the canonical drid for that case, English preferred). Useful when you want a unique-cases index.

Details

- see exactly which drids ship as known-valid (no need to fetch to find out);
- subset to English-only or French-only case lists without running the full harvester;
- map between drid (URL fragment) and case_number (SIU's own identifier) offline.

The manifest is refreshed by maintainers via `morie_siu_refresh_manifest()`; it ships gzipped under `inst/extdata/` at ~50 KB.

Value

A data frame with columns `drid`, `http_code`, `body_bytes`, `attempts`, `case_number`, `_language`, `source`, `retrieved_at_utc`, `canonical_drid`.

Examples

```
idx <- morie_siu_index(lang = "en")
head(idx)
# How many drids are English vs French vs unknown?
table(morie_siu_index()$`_language`)
# Unique-case index (English-preferred)
canon <- morie_siu_index(canonical_only = TRUE)
nrow(canon)
```

`morie_siu_llm_extract` *Extract SIU report fields with an LLM (Gemini or Claude)*

Description

Sends the cached director's-report HTML for one case through a large-language-model endpoint and asks it to return the 64-column morie schema as JSON. The result is in the SAME row format as the C++ parser, so it drops straight into `morie_siu_compare()` as the `external` argument for an independent diff against the parser.

Usage

```
morie_siu_llm_extract(
  case_number,
  model = c("ollama", "gemini"),
  cache_dir = file.path(tempdir(), "morie", "siu"),
  max_html_chars = 80000L,
  mock_response_text = NULL
)
```

Arguments

`case_number`	An SIU case number (e.g. `"17-OVI-201"`).
`model`	One of `"ollama"` (default; free, runs locally, zero-config when an Ollama daemon is on `localhost:11434`), `"gemini"` (paid), or `"claude"` (paid). A character vector enables fail-over: the first model whose call succeeds wins. The default `c("ollama", "gemini")` tries the local free model first and only escalates to paid Gemini if Ollama isn't installed or fails – so morie costs $0 to use as long as you have a free Gemma / Qwen / Llama running locally (e.g. `ollama pull gemma3:4b`).
`cache_dir`	Directory holding the harvester's SIU.csv and the optional `html/` subdirectory.
`max_html_chars`	Soft cap on the HTML payload sent to the model (default 80,000 – larger than any real SIU report, small enough to stay under typical context budgets).
`mock_response_text`	For testing only: if non-NULL, skip the network call and use this string as the model's raw reply.

Details

The cached HTML remains the ground truth. This function does not claim the LLM is more accurate than the regex parser; it provides a fast second extraction so disagreements between two independent methods (regex vs. LLM) can be flagged for human review against the saved report.

Credentials are read from environment variables only – never hard-coded, never passed as function arguments – so secrets do not leak into call traces, logs, or scripts. Set `GOOGLE_API_KEY` for Gemini, `ANTHROPIC_API_KEY` for Claude, or `OLLAMA_HOST` (e.g. `"http://localhost:11434"` or an OllamaFreeAPI base URL) plus optionally `OLLAMA_MODEL` (default `"llama3.2:3b"`) for Ollama-compatible open-weight endpoints.

Value

A one-row data frame with the 64 morie SIU columns. Any field the model could not extract is the empty string (matching the C++ parser's convention).

Examples

```
## Not run:
Sys.setenv(GOOGLE_API_KEY = "your-gemini-key")
r <- morie_siu_llm_extract("17-OVI-201", model = "gemini")
# Diff parser vs LLM against the HTML:
morie_siu_compare(
  "17-OVI-201",
  external = r,
  field_map = setNames(as.list(names(r)), names(r)),
  external_case_col = "case_number"
)

## End(Not run)
```

`morie_siu_mental_health_race_indicators` *Mental-health / race keyword indicators*

Description

Frequency table of the semicolon-delimited keyword tags written by the parser into `mental_health_or_race_indic`
The parser scans narratives against a closed vocabulary (see `morie.siu._parser._MH_RACE_KEYWORDS`);
this analysis tallies those tags across the case corpus.

Usage

```
morie_siu_mental_health_race_indicators(data = NULL)
```

Arguments

`data`	Either a data frame (e.g. the output of `morie_fetch_siu()` read in) or a path to `SIU_by_case.csv`. `NULL` (the default) looks in `file.path(tempdir(), "morie", "siu", "SIU_by_case.csv")`.

Value

A `morie_rich_result` list, including a warning noting that keyword-presence is a signal, not a verdict.

`morie_siu_parse_html` *Parse a SIU director's-report HTML page (pure-R)*

Description

Parse a SIU director's-report HTML page (pure-R)

Usage

```
morie_siu_parse_html(html, drid = NA_integer_, source_url = NULL)
```

Arguments

`html`	Raw HTML response body.
`drid`	Optional integer drid from the request URL – useful when the page itself doesn't echo it.
`source_url`	Optional canonical URL – used to derive drid and recorded as `source_url_report`.

Value

A list with every `SIU_COLUMNS` key (NAs for unfound fields).

`morie_siu_parse_news_html`
Parse a SIU news-release HTML page (pure-R)

Description

News-release pages live at `news_template.php?nrid=<N>` and have a different layout than the director's reports – a single headline, short summary paragraph, signed-by-Director line.

Usage

```
morie_siu_parse_news_html(html, nrid = NA_integer_, source_url = NULL)
```

Arguments

`html`	Raw HTML response body.
`nrid`	Optional integer nrid from the request URL.
`source_url`	Optional canonical URL.

Value

A list with `nrid`, `source_url_news`, `news_release_title`, `news_release_date_iso`, `news_release_date_raw`, `news_release_summary`, `case_number`, and `directors_name`.

`morie_siu_parser` *Pure-R SIU director's-report parser (port of* `morie.siu._parser`*)*

Description

Parses one SIU director's-report HTML page (or one news-release page) into a structured row list. The production parser lives in the Rcpp / C++ backend (`.siu_parse_report`, `.siu_parse_news`); this pure-R port is provided as a reference implementation and as a fallback for environments where the compiled **libmorie** backend is unavailable.

Details

Suggested dependencies. These functions optionally use **rvest** + **xml2** for DOM walking; without them, a regex-based fallback over flat tag-stripped text is used. Either way the parser is pure (no network) – hand it a raw HTML string and it returns a row dict matching `SIU_COLUMNS`.

Hardened against the SIU page markup shifting over time by:

- looking for several label variants per field,
- falling back to regex on stripped text when DOM structure shifts,
- preserving the verbatim `narrative_full` regardless of parse success.

`morie_siu_record_correction`
Record a verified correction to the SIU parser's output

Description

Saves a (case_number, field, verified_value) tuple to a local overrides CSV at `<cache_dir>/canonical_overrides` Every subsequent `morie_fetch_siu()` on that `cache_dir` will overlay these corrections onto the regex-parsed output. The shipped `inst/extdata/siu_canonical_overrides.csv.gz` carries maintainer-confirmed corrections; this function lets users add their own without touching the package source.

Usage

```
morie_siu_record_correction(
  case_number,
  field,
  verified_value,
  note = "",
  cache_dir = file.path(tempdir(), "morie", "siu")
)
```

Arguments

`case_number`	SIU case number, e.g. `"17-OVI-201"`.
`field`	Name of the column in the SIU schema (e.g. `"location_of_call"`).
`verified_value`	The correct value, verified against the cached HTML (see `morie_siu_audit_case()`).
`note`	Optional one-line note describing the basis for the correction (HTML excerpt, LLM verdict, etc.).
`cache_dir`	Directory holding the harvester's SIU.csv.

Details

This is the "memory" of the parser: every wrong cell you find and fix becomes permanent for that cache_dir. Maintainers can submit corrections upstream by sharing the resulting CSV file.

Value

Invisibly, the path to the updated overrides CSV.

Examples

```
# Writes the correction to a temp cache so the example never
# touches the per-user cache directory.
tmp <- tempfile("morie_siu_"); dir.create(tmp, recursive = TRUE)
morie_siu_record_correction(
  case_number = "17-OVI-201",
  field = "location_of_call",
  verified_value = "Clair Road East, City of Guelph",
  note = "HTML excerpt: 'on Clair Road East in the City of Guelph'",
  cache_dir = tmp
)
unlink(tmp, recursive = TRUE)
```

`morie_siu_refresh_manifest` *Rebuild the Ontario SIU DRID manifest by probing the live site*

Description

Sweeps director's-report ids `1..max_drid` and writes a small CSV recording which ids return a healthy report page, the parsed case number, and the response body size. The harvester (`morie_fetch_siu`) then uses this manifest to short-circuit the ~30-50 percent of ids that have no report, saving bandwidth and WAF-trigger risk on every run.

Usage

```
morie_siu_refresh_manifest(
  out_path = NULL,
  max_drid = NULL,
  min_drid = 1L,
  concurrency = 4L,
  rate_rps = 4,
  progress = TRUE
)
```

Arguments

`out_path`	Path to write the gzipped CSV. Default is the in-place manifest location (only useful for maintainers building from a source checkout).
`max_drid`	Highest drid to probe. Default `NULL` auto-discovers from the SIU index endpoint and adds a margin.
`min_drid`	Lowest drid to probe (default `1L`).
`concurrency`	Maximum simultaneous transfers (default `4`).
`rate_rps`	Maximum request starts per second (default `4`).
`progress`	Logical; print a per-batch progress line.

Details

The shipped manifest at `inst/extdata/siu_drid_manifest.csv.gz` is a snapshot. Users who want the latest can call this function; it is also how morie maintainers regenerate the snapshot.

Value

Invisibly, a data frame of the full sweep (every probed drid, including misses), parallel to what was written to `out_path`.

Examples

```
## Not run:
# Network: refreshes the manifest by probing the SIU site
# (~25-40 min at the default polite rate of 4 RPS for ~6000 ids).
df <- morie_siu_refresh_manifest(out_path = tempfile(fileext = ".csv.gz"))
table(df$http_code)

## End(Not run)
```

`morie_siu_sanity_check` *Row-level sanity check on a parsed SIU table (regex-only, no LLM)*

Description

For every row in a parser-emitted SIU table, flag cells that don't match the expected format for their column – `case_number` that doesn't look like an SIU case id, `date_*_iso` that isn't a valid ISO 8601 date, `number_of_*` that isn't a positive integer, `charges_recommended` that isn't "Yes" / "No", etc. Returns a data frame ranked by issue count so the most-broken rows surface at the top for manual inspection against the cached HTML.

Usage

```
morie_siu_sanity_check(df)
```

Arguments

`df`	A data frame in the morie SIU 64-column schema, or a path to such a CSV.

Details

Designed to be a fast first-pass quality filter – runs in milliseconds, no network, no LLM, no API key. Doesn't try to verify correctness against the underlying report (that's what `morie_siu_audit_columns()` is for); just checks that each value MATCHES THE EXPECTED FORMAT for its field. A clean sanity check is necessary but not sufficient for correctness.

Value

A data frame with one row per source row, columns: `case_number`, `drid`, `issues_count` (integer number of suspicious cells), `issues` (semicolon-separated string of `field:reason` pairs). Ordered descending by `issues_count`.

Examples

```
## Not run:
csv <- morie_fetch_siu(cache_dir = tempdir(), cache_html = TRUE)
sanity <- morie_siu_sanity_check(csv)
head(sanity, 10) # worst 10 rows -- inspect against HTML
table(sanity$issues_count)

## End(Not run)
```

`morie_siu_sprott_doob_feb2021`
Comprehensive replication of Sprott & Doob (Feb 2021)

Description

Bundles Tables 13, 19, and 23 (the three headline tables) into a single `morie_siu_result` with cross-references.

Usage

```
morie_siu_sprott_doob_feb2021()
```

Value

A `morie_siu_result`.

Examples

```
morie_siu_sprott_doob_feb2021()$payload$headline_findings$n_total_stays
```

`morie_siu_sprott_doob_iftene_table1`
Sprott-Doob-Iftene (May 2021) Table 1: IEDM-reviewed population

---

Description

Sprott-Doob-Iftene (May 2021) Table 1: IEDM-reviewed population

Usage

```
morie_siu_sprott_doob_iftene_table1()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_iftene_table10` *Sprott-Doob-Iftene (May 2021) Table 10: per-IEDM decision variance*

Description

Sprott-Doob-Iftene (May 2021) Table 10: per-IEDM decision variance

Usage

```
morie_siu_sprott_doob_iftene_table10()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_iftene_table15` *Sprott-Doob-Iftene (May 2021) Table 15: long-stay no-IEDM cases*

Description

Sprott-Doob-Iftene (May 2021) Table 15: long-stay no-IEDM cases

Usage

```
morie_siu_sprott_doob_iftene_table15()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_iftene_table9`
Sprott-Doob-Iftene (May 2021) Table 9: IEDM review outcomes

Description

Sprott-Doob-Iftene (May 2021) Table 9: IEDM review outcomes

Usage

```
morie_siu_sprott_doob_iftene_table9()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_table11`
Sprott-Doob (Feb 2021) Table 11: Region x stay length

Description

Sprott-Doob (Feb 2021) Table 11: Region x stay length

Usage

```
morie_siu_sprott_doob_table11()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_table12`
Sprott-Doob (Feb 2021) Table 12: regional over-/under-representation

---

Description

Sprott-Doob (Feb 2021) Table 12: regional over-/under-representation

Usage

```
morie_siu_sprott_doob_table12()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_table13`
Sprott-Doob (Feb 2021) Table 13: regional SIU person-stay rates

---

Description

Sprott-Doob (Feb 2021) Table 13: regional SIU person-stay rates

Usage

```
morie_siu_sprott_doob_table13()
```

Value

A `morie_siu_result` (named list with the replicated table, summary lines, interpretation, and payload).

Examples

```
morie_siu_sprott_doob_table13()$payload$qc_on_short_stay_ratio
```

`morie_siu_sprott_doob_table15`
Sprott-Doob (Feb 2021) Table 15: Region x MH-flag

Description

Sprott-Doob (Feb 2021) Table 15: Region x MH-flag

Usage

```
morie_siu_sprott_doob_table15()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_table19`
Sprott-Doob (Feb 2021) Table 19: Mandela-Rules classification

---

Description

Sprott-Doob (Feb 2021) Table 19: Mandela-Rules classification

Usage

```
morie_siu_sprott_doob_table19()
```

Value

A `morie_siu_result`.

Examples

```
morie_siu_sprott_doob_table19()$payload$pct_problematic
```

`morie_siu_sprott_doob_table22`
Sprott-Doob (Feb 2021) Table 22: Region x Mandela groups

Description

Sprott-Doob (Feb 2021) Table 22: Region x Mandela groups

Usage

```
morie_siu_sprott_doob_table22()
```

Value

A `morie_siu_result`.

`morie_siu_sprott_doob_table23`
Sprott-Doob (Feb 2021) Table 23: regional torture/solitary rates

---

Description

Sprott-Doob (Feb 2021) Table 23: regional torture/solitary rates

Usage

```
morie_siu_sprott_doob_table23()
```

Value

A `morie_siu_result`.

Examples

```
morie_siu_sprott_doob_table23()$payload$pac_on_torture_ratio
```

`morie_siu_sprott_doob_table4`
Sprott-Doob (Feb 2021) Table 4: length-of-stay distribution

Description

Sprott-Doob (Feb 2021) Table 4: length-of-stay distribution

Usage

```
morie_siu_sprott_doob_table4()
```

Value

A `morie_siu_result`.

`morie_siu_translate` *Translate SIU report text into any target language via local LLM*

Description

For SIU cases whose parser-emitted text isn't in the reader's preferred language, translate the long-form text fields into `target_lang` via a local Ollama model (default $0 cost, no API key) and save each translation as a canonical override. Subsequent `morie_fetch_siu()` runs then return text in `target_lang` for those cases automatically.

`morie_siu_translate_fr_to_en` is a thin back-compat wrapper that calls `morie_siu_translate` with `target_lang = "en"`, `source_lang = "fr"`.

Usage

```
morie_siu_translate(
  target_lang = NULL,
  source_lang = NULL,
  case_numbers = NULL,
  model = "ollama",
  fields = c("narrative_summary", "news_release_summary", "news_release_title",
    "relevant_legislation"),
  cache_dir = file.path(tempdir(), "morie", "siu"),
  progress = TRUE
)

morie_siu_translate_fr_to_en(
  case_numbers = NULL,
  model = "ollama",
  fields = c("narrative_summary", "news_release_summary", "news_release_title",
    "relevant_legislation"),
  cache_dir = file.path(tempdir(), "morie", "siu"),
  progress = TRUE
)
```

Arguments

`target_lang`	Target ISO 639-1 language code (or full language name). Defaults to `Sys.getenv("MORIE_USER_LANG")` or, failing that, the first two characters of `Sys.getenv("LANG")` – so it picks up the user's system locale automatically.
`source_lang`	Source language code, or `NULL` (default) to use each row's parsed `_language` field.
`case_numbers`	Character vector of SIU case numbers to translate. Defaults to every row whose `_language` differs from `target_lang` and has no override yet.
`model`	LLM model chain (see `morie_siu_llm_extract`). Default `"ollama"` for $0 cost via local Gemma / etc.

`fields`	Which text fields to translate. Defaults to the long-form fields that benefit most from translation: `narrative_summary`, `news_release_summary`, `news_release_title`, `relevant_legislation`.
`cache_dir`	Directory holding the harvester's SIU.csv and cached HTML.
`progress`	Print per-case progress.

Details

Use cases:

- French-only SIU reports (a few per year of SIU output) that have no English-paired drid – translate to "en" so downstream analyses can join them with the rest.
- English SIU reports that a Hindi / Spanish / Mandarin / Punjabi / Arabic / etc. reader needs – translate to their first language for accessibility.
- Any cross-language pivot for community-oriented publication, where the reader's first language isn't what the SIU originally published in.

Idempotent (skips cases that already have an override on file for this `target_lang`). Self-improving (every translation accumulates in `<cache_dir>/canonical_overrides.csv`, so the SIU table becomes more accessible every time you run this). Maintainers can promote the resulting overrides into the shipped `inst/extdata/siu_canonical_overrides.csv.gz`.

For best speed/quality on multilingual translation use `OLLAMA_MODEL=translategemma:latest` – a Gemma model fine-tuned for translation. Falls back to whatever model `OLLAMA_MODEL` points at.

Value

Invisibly, a data frame of newly-recorded (case_number, field, verified_value) translations.

Examples

```
## Not run:
Sys.setenv(
  OLLAMA_HOST = "http://localhost:11434",
  OLLAMA_MODEL = "translategemma:latest"
)
csv <- morie_fetch_siu(cache_html = TRUE)
# Translate every non-English row to English:
morie_siu_translate(target_lang = "en")
# Or translate everything to Hindi for a Hindi-first reader:
morie_siu_translate(target_lang = "hi")
# Re-fetch picks up the new overrides automatically:
csv <- morie_fetch_siu(overwrite = TRUE)

## End(Not run)
```

`morie_siu_verify_chi2`
Recompute Pearson chi-square from a 2D contingency table

Description

Pure-base-R Pearson chi-square without Yates correction. Intended for quick self-checks of the transcribed cell counts against the published chi-square values.

Usage

```
morie_siu_verify_chi2(observed)
```

Arguments

`observed`	A 2D matrix or data frame of non-negative counts.

Value

A named list with elements `chi2`, `df`, `p_value`, `expected`, and `n`.

Examples

```
morie_siu_verify_chi2(matrix(c(10, 10, 10, 10), nrow = 2))$chi2
```

`morie_siu_verify_published_chi_squares`
Verify every published chi-square against its transcribed cells

---

Description

Cross-checks Sprott-Doob Tables 11, 15, 22 (Feb 2021) and Sprott- Doob-Iftene Tables 5, 10 (May 2021) by recomputing the chi-square from each transcribed contingency table and comparing it to the published value. A ”pass” means the recomputed chi-square is within rounding tolerance (1.0-1.5 units) of the published value.

Usage

```
morie_siu_verify_published_chi_squares()
```

Value

A `morie_siu_result` with the verification table and per-table warnings for any mismatch.

Examples

```
v <- morie_siu_verify_published_chi_squares()
v$payload$n_pass
```

`MORIE_SIUIAP_AFFIDAVITS` *Federal Court affidavits / expert evidence indexed by* `morie`.

Description

Federal Court affidavits / expert evidence indexed by `morie`.

Usage

```
MORIE_SIUIAP_AFFIDAVITS
```

Format

An object of class `list` of length 1.

`morie_siuiap_cite` *Build a citation string for an SIU IAP / CRIMSL / affidavit entry.*

Description

Searches `MORIE_SIUIAP_REPORTS`, then `MORIE_SIUIAP_CRIMSL_REPORTS`, then `MORIE_SIUIAP_AFFIDAVITS`, in order, and returns a one-line citation in the form `<authors> (<year>). <title>. <publisher>.`.

Usage

```
morie_siuiap_cite(report_id = "final_2024")
```

Arguments

`report_id`	Character scalar. One of the names of `MORIE_SIUIAP_REPORTS`, `MORIE_SIUIAP_CRIMSL_REPORTS`, or `MORIE_SIUIAP_AFFIDAVITS`. Defaults to `"final_2024"`.

Value

A character scalar citation. Errors on unknown `report_id`.

Examples

```
morie_siuiap_cite("final_2024")
morie_siuiap_cite("sprott_doob_torture_solitary_2021")
```

`MORIE_SIUIAP_CRIMSL_REPORTS`
CRIMSL UToronto Sprott / Doob / Iftene research reports (2020-2021).

Description

CRIMSL UToronto Sprott / Doob / Iftene research reports (2020-2021).

Usage

```
MORIE_SIUIAP_CRIMSL_REPORTS
```

Format

An object of class `list` of length 4.

`MORIE_SIUIAP_ORIGINAL_PANEL_2019_2020`
Earlier (Doob-chaired) panel, established 2019, dissolved mid-2020.

Description

Earlier (Doob-chaired) panel, established 2019, dissolved mid-2020.

Usage

```
MORIE_SIUIAP_ORIGINAL_PANEL_2019_2020
```

Format

An object of class `list` of length 6.

`MORIE_SIUIAP_PANEL_MANDATE`
SIU IAP panel mandate (long-form prose).

Description

SIU IAP panel mandate (long-form prose).

Usage

```
MORIE_SIUIAP_PANEL_MANDATE
```

Format

An object of class `character` of length 1.

`MORIE_SIUIAP_PANEL_MEMBERS`
SIU IAP panel members (2021-2024 panel, Sapers-chaired).

Description

SIU IAP panel members (2021-2024 panel, Sapers-chaired).

Usage

```
MORIE_SIUIAP_PANEL_MEMBERS
```

Format

An object of class `list` of length 3.

`morie_siuiap_panel_summary`
Human-readable summary of the SIU IAP panel.

Description

Human-readable summary of the SIU IAP panel.

Usage

```
morie_siuiap_panel_summary()
```

Value

A character scalar summarising chair, members, mandate dates, and the Public Safety Canada URL.

Examples

```
cat(morie_siuiap_panel_summary())
```

MORIE_SIUIAP_REPORTS *SIU IAP panel reports (Public Safety Canada, 2022-2024).*

Description

SIU IAP panel reports (Public Safety Canada, 2022-2024).

Usage

```
MORIE_SIUIAP_REPORTS
```

Format

An object of class `list` of length 5.

MORIE_SIUIAP_URL *SIU IAP – Public Safety Canada landing page URL.*

Description

SIU IAP – Public Safety Canada landing page URL.

Usage

```
MORIE_SIUIAP_URL
```

Format

An object of class `character` of length 1.

`morie_spatial_voting_aldrich_mckelvey` *Aldrich-McKelvey scaling*

Description

Recovers latent stimulus positions from perceptual placement data by estimating respondent-specific intercepts a_i and slopes b_i in the model

$$z_{ij} = a_i + b_i \hat{z}_j + \epsilon_{ij}.$$

Delegates to `basicspace::aldmck` when the `basicspace` package is installed; otherwise a hand-rolled EM/least-squares fallback is used.

Usage

```
morie_spatial_voting_aldrich_mckelvey(
  Z,
  n_dims = 1L,
  max_iter = 100L,
  tol = 1e-06
)
```

Arguments

`Z`	A respondent-by-stimulus numeric matrix of perceptual placements. `NA` entries are treated as missing.
`n_dims`	Number of latent dimensions (typically 1).
`max_iter`	Maximum EM iterations for the fallback solver.
`tol`	Convergence tolerance on the stimulus configuration.

Value

A list with components `zhat` (stimulus positions), `alpha`, `beta`, `weights`, `iterations`, `converged`, and `engine` ("basicspace" or "fallback").

References

Aldrich, J. H. and McKelvey, R. D. (1977). "A Method of Scaling with Applications to the 1968 and 1972 Presidential Elections." *American Political Science Review*, 71(1), 111-130.

Poole, K. T. (1998). "Recovering a Basic Space from a Set of Issue Scales." *American Journal of Political Science*, 42(3), 954-993.

Armstrong, D. A., Bakker, R., Carroll, R., Hare, C., Poole, K. T., and Rosenthal, H. (2021). *Analyzing Spatial Models of Choice and Judgment*, 2nd ed. Chapman & Hall/CRC.

Examples

```
set.seed(1)
Z <- matrix(rnorm(20 * 5), 20, 5)
fit <- morie_spatial_voting_aldrich_mckelvey(Z)
fit$zhat
```

---

morie_spatial_voting_alpha_nominate

Alpha-NOMINATE (stub)

Description

Carroll et al. (2013) mixture model between Gaussian and quadratic utility, sampled via slice sampling (Neal 2003). Porting the slice sampler is beyond this session's budget.

Usage

```
morie_spatial_voting_alpha_nominate(
  votes,
  n_dims = 2L,
  n_samples = 500L,
  burn_in = 100L,
  seed = 42L
)
```

Arguments

`votes`	Vote matrix. @param n_dims Latent dimensions.
`n_dims`	Integer; latent ideal-point dimensionality (default 2).
`n_samples`	MCMC samples. @param burn_in Burn-in length.
`burn_in`	Integer; MCMC burn-in iterations to discard before summarising the posterior.
`seed`	RNG seed.

Value

Never returns; raises `NotYetPorted`.

References

Carroll, R., Lewis, J. B., Lo, J., Poole, K. T., and Rosenthal, H. (2013); Neal, R. M. (2003) *Annals of Statistics*.

Examples

```
## Not run: morie_spatial_voting_alpha_nominate(matrix(0, 5, 5))
```

morie_spatial_voting_anchoring_vignettes

Anchoring vignettes for DIF correction

Description

Anchoring vignettes for DIF correction

Usage

```
morie_spatial_voting_anchoring_vignettes(Y, V, n_categories = 5L)
```

Arguments

`Y`	Vector of self-placement ratings.
`V`	Respondent-by-vignette ratings.
`n_categories`	Number of ordered categories.

Value

List with `corrected_scores`, `thresholds`, `dif_estimates`, `vignette_order`, `n_respondents`, `n_vignettes`.

References

King, G., Murray, C. J. L., Salomon, J. A., and Tandon, A. (2003). "Enhancing the Validity and Cross-Cultural Comparability of Measurement in Survey Research." *APSR*, 97(4), 567-583.

Examples

```
Y <- sample(1:5, 30, replace = TRUE)
V <- matrix(sample(1:5, 30 * 3, replace = TRUE), 30, 3)
morie_spatial_voting_anchoring_vignettes(Y, V)
```

morie_spatial_voting_bayesian_am

Bayesian Aldrich-McKelvey scaling (stub)

Description

Bayesian Aldrich-McKelvey scaling (stub)

Usage

```
morie_spatial_voting_bayesian_am(
  Z,
  n_samples = 1000L,
  burn_in = 200L,
  prior_sd = 10
)
```

Arguments

`Z`	Perceptual placement matrix.
`n_samples`	MCMC samples.
`burn_in`	Burn-in length.
`prior_sd`	Prior SD on stimulus positions.

Value

Never returns; raises `NotYetPorted`.

References

Hare, C., Armstrong, D. A., Bakker, R., Carroll, R., and Poole, K. T. (2015). "Using Bayesian Aldrich-McKelvey Scaling to Study Citizens' Ideological Preferences and Perceptions." *AJPS*, 59(3).

Examples

```
## Not run: morie_spatial_voting_bayesian_am(matrix(rnorm(50), 10, 5))
```

`morie_spatial_voting_bayesian_irt_likelihood`

Bayesian IRT likelihood (deterministic part of CJR machinery)

Description

Bayesian IRT likelihood (deterministic part of CJR machinery)

Usage

```
morie_spatial_voting_bayesian_irt_likelihood(votes, x, alpha, beta)
```

Arguments

`votes`	Binary matrix. @param x Ideal points.
`x`	Matrix or data.frame of vote data (rows = legislators, columns = roll-call votes).
`alpha`	Difficulty. @param beta Discrimination.
`beta`	Numeric vector of item-difficulty parameters; one entry per column of `x`.

Value

List with `loglik`, `vote_probs`, `n_correct`, `n_total`, `accuracy`.

References

Clinton, Jackman & Rivers (2004).

Examples

```
v <- matrix(stats::rbinom(20, 1, 0.5), 4, 5)
morie_spatial_voting_bayesian_irt_likelihood(
  v, matrix(rnorm(4), 4, 1), rep(0, 5), matrix(rnorm(5), 5, 1))
```

`morie_spatial_voting_bayesian_irt_posterior` *Posterior summaries for a Bayesian IRT chain*

Description

Posterior summaries for a Bayesian IRT chain

Usage

```
morie_spatial_voting_bayesian_irt_posterior(chain, standardize = TRUE)
```

Arguments

`chain`	Array of shape (n_samples, n_leg, n_dims).
`standardize`	Whether to per-sample standardise.

Value

List with `posterior_mean`, `posterior_sd`, `ci_lower`, `ci_upper`, `n_samples`, `standardized`.

References

Jackman (2009).

Examples

```
ch <- array(rnorm(100 * 5 * 2), c(100, 5, 2))
morie_spatial_voting_bayesian_irt_posterior(ch)
```

`morie_spatial_voting_bayesian_mds` *Bayesian MDS (stub) – log-normal distances via Metropolis*

Description

Bayesian MDS (stub) – log-normal distances via Metropolis

Usage

```
morie_spatial_voting_bayesian_mds(
  D,
  n_dims = 2L,
  n_samples = 1000L,
  burn_in = 200L,
  sigma_init = 1
)
```

Arguments

`D`	Distance matrix. @param n_dims Dimensions. @param n_samples MCMC samples.
`n_dims`	Integer; latent dimensionality.
`n_samples`	Integer; posterior-sample count.
`burn_in`	Burn-in length. @param sigma_init Initial sigma.
`sigma_init`	Numeric; initial value for the latent-coordinate scale (default 1).

Value

Never returns; raises `NotYetPorted`.

References

Oh & Raftery (2001) JASA 96(455).

Examples

```
## Not run: morie_spatial_voting_bayesian_mds(matrix(0, 5, 5))
```

`morie_spatial_voting_bayesian_unfolding`
Bayesian unfolding (stub) – Bakker & Poole sampler

Description

Bayesian unfolding (stub) – Bakker & Poole sampler

Usage

```
morie_spatial_voting_bayesian_unfolding(
  D,
  n_dims = 2L,
  n_samples = 1000L,
  burn_in = 200L
)
```

Arguments

`D`	Respondent-stimulus dissimilarity matrix.
`n_dims`	Latent dimensions. @param n_samples MCMC samples.
`n_samples`	Integer; posterior-sample count.
`burn_in`	Burn-in length.

Value

Never returns; raises `NotYetPorted`.

References

Bakker, R. and Poole, K. T. (2013).

Examples

```
## Not run: morie_spatial_voting_bayesian_unfolding(matrix(0, 3, 4))
```

`morie_spatial_voting_blackbox`
Blackbox / Basic Space scaling

Description

Recovers respondent ideal points from an issue-scale response matrix via SVD on the column-centred matrix. Implements Poole's (1998) decomposition $X_0 = \Psi W' + J_n c' + E_0$. Delegates to `basicspace::blackbox` when available.

Usage

```
morie_spatial_voting_blackbox(X, n_dims = 2L)
```

Arguments

X	A respondent-by-issue numeric matrix of responses (NA for missing).
n_dims	Number of dimensions to extract.

Value

A list with ideal_points, stimuli_weights, eigenvalues, singular_values, explained_variance, col_means, n_dims, and engine.

References

Poole, K. T. (1998); Armstrong et al. (2021).

Examples

```
set.seed(1)
X <- matrix(rnorm(30 * 6), 30, 6)
morie_spatial_voting_blackbox(X, n_dims = 2)
```

---

morie_spatial_voting_cjr_irt

Clinton-Jackman-Rivers Bayesian IRT (stub)

Description

Clinton-Jackman-Rivers Bayesian IRT (stub)

Usage

```
morie_spatial_voting_cjr_irt(
  votes,
  n_dims = 1L,
  n_samples = 1000L,
  burn_in = 200L
)
```

Arguments

votes	Binary roll-call matrix.
n_dims	Ideal-point dimensions.
n_samples	MCMC samples. @param burn_in Burn-in length.
burn_in	Integer; MCMC burn-in iterations.

Value

Never returns; raises `NotYetPorted`.

References

Clinton, Jackman & Rivers (2004).

Examples

```
## Not run: morie_spatial_voting_cjr_irt(matrix(0, 5, 5))
```

`morie_spatial_voting_classical_mds` *Classical (metric) multidimensional scaling*

Description

Torgerson scaling via eigendecomposition of the double-centred matrix.

Usage

```
morie_spatial_voting_classical_mds(D, n_dims = 2L)
```

Arguments

`D`	Symmetric numeric distance matrix.
`n_dims`	Number of dimensions to extract.

Value

A list with `coordinates`, `eigenvalues`, `stress`, `fit`, `B_matrix`.

References

Torgerson, W. S. (1952); Armstrong et al. (2021).

Examples

```
D <- as.matrix(dist(matrix(rnorm(40), 10)))
morie_spatial_voting_classical_mds(D, n_dims = 2)
```

`morie_spatial_voting_cutting_lines`
Cutting-line endpoints for Coombs-mesh plots

Description

Cutting-line endpoints for Coombs-mesh plots

Usage

```
morie_spatial_voting_cutting_lines(normals, cutpoints, xlim = c(-1, 1))
```

Arguments

`normals`	(n_votes x n_dims) normal vectors.
`cutpoints`	Numeric cutpoint offsets.
`xlim`	Length-2 numeric vector of x-axis limits.

Value

List with `endpoints` (list of pairs), `angles`, `midpoints`, `n_lines`.

References

Poole (2005).

Examples

```
morie_spatial_voting_cutting_lines(matrix(rnorm(6), 3, 2), c(0.1, -0.2, 0))
```

`morie_spatial_voting_double_centering`
Double-center a distance matrix

Description

Computes $B = -\frac{1}{2} H D^{(2)} H$ with $H = I - n^{-1}\mathbf{1}\mathbf{1}'$.

Usage

```
morie_spatial_voting_double_centering(D)
```

Arguments

`D`	Symmetric numeric distance matrix.

Value

The double-centered matrix B.

References

Torgerson (1952); Armstrong et al. (2021), Section 3.

Examples

```
morie_spatial_voting_double_centering(as.matrix(dist(matrix(rnorm(20), 5))))
```

---

morie_spatial_voting_dw_nominate

DW-NOMINATE dynamic weighted ideal-point estimator

Description

Gaussian-error NOMINATE variant supporting comparable scores across legislative sessions.

Usage

```
morie_spatial_voting_dw_nominate(
  votes,
  n_dims = 2L,
  max_iter = 100L,
  tol = 1e-06
)
```

Arguments

`votes`	Legislator-by-vote binary matrix.
`n_dims`	Latent dimensions.
`max_iter`	Maximum iterations.
`tol`	Tolerance (unused; kept for API parity).

Value

List with `ideal_points`, `dim_weights`, `normal_vectors`, `cutpoints`, `log_lik`, `gmp`, `n_dims`.

References

Poole, K. T. and Rosenthal, H. (1997). *Congress: A Political- Economic History of Roll Call Voting*. Oxford University Press.

Examples

```
set.seed(1)
v <- matrix(stats::rbinom(20 * 30, 1, 0.5), 20, 30)
morie_spatial_voting_dw_nominate(v, max_iter = 20)
```

---

morie_spatial_voting_dynamic_irt

Dynamic IRT with random-walk priors (stub)

Description

Time-series IRT where ideal points evolve via a random walk: $\phi_{i,t} \sim N(\phi_{i,t-1}, \tau^2)$.

Usage

```
morie_spatial_voting_dynamic_irt(
  votes,
  time_periods,
  n_samples = 500L,
  burn_in = 100L,
  seed = 42L
)
```

Arguments

`votes`	Vote matrix. @param time_periods Per-vote period indices.
`time_periods`	Integer vector of period indices (one per roll call) for the dynamic-IRT random-walk prior on ideal points.
`n_samples`	MCMC samples. @param burn_in Burn-in length.
`burn_in`	Integer; MCMC burn-in iterations.
`seed`	RNG seed.

Value

Never returns; raises `NotYetPorted`.

References

Martin, A. D. and Quinn, K. M. (2002). "Dynamic Ideal Point Estimation via Markov Chain Monte Carlo for the U.S. Supreme Court, 1953-1999." *Political Analysis*, 10(2).

Examples

```
## Not run: morie_spatial_voting_dynamic_irt(matrix(0, 4, 4), 1:4)
```

morie_spatial_voting_em_irt

EM algorithm for binary IRT

Description

Imai, Lo & Olmsted (2016) closed-form EM updates for binary IRT, suitable for very large vote matrices where MCMC is infeasible.

Usage

```
morie_spatial_voting_em_irt(votes, n_dims = 1L, max_iter = 100L, tol = 1e-06)
```

Arguments

votes	Legislator-by-vote binary matrix.
n_dims	Latent dimensions.
max_iter	Maximum EM iterations.
tol	Convergence tolerance on ideal-point change.

Value

List with ideal_points, discrimination, difficulty, log_lik, iterations.

References

Imai, K., Lo, J., and Olmsted, J. (2016). "Fast Estimation of Ideal Points with Massive Data." *APSR*, 110(4), 631-656.

Examples

```
set.seed(1)
v <- matrix(stats::rbinom(20 * 30, 1, 0.5), 20, 30)
morie_spatial_voting_em_irt(v, max_iter = 20L)
```

morie_spatial_voting_ideal_point_recovery

Ideal-point recovery from unfolding output

Description

Ideal-point recovery from unfolding output

Usage

```
morie_spatial_voting_ideal_point_recovery(X_r, X_s = NULL)
```

Arguments

X_r	Respondent coordinates.
X_s	Stimulus coordinates (unused; the respondent row IS the ideal point).

Value

Numeric matrix of respondent ideal points.

References

Armstrong et al. (2021), Section 4.5.

Examples

```
morie_spatial_voting_ideal_point_recovery(matrix(rnorm(6), 3, 2))
```

---

morie_spatial_voting_indscal

INDSCAL: weighted MDS with individual differences

Description

Carroll & Chang (1970) weighted MDS with a shared stimulus space and per-individual dimension weights.

Usage

```
morie_spatial_voting_indscal(
  dissimilarities,
  n_dims = 2L,
  max_iter = 300L,
  tol = 1e-06
)
```

Arguments

dissimilarities	List of (n_stim x n_stim) dissimilarity matrices.
n_dims	Number of dimensions.
max_iter	Maximum ALS iterations.
tol	Convergence tolerance on configuration change.

Value

List with `group_config`, `weights`, `stress`, `iterations`, `n_individuals`, `n_stimuli`.

References

Carroll, J. D. and Chang, J.-J. (1970). "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of Eckart-Young Decomposition." *Psychometrika*, 35(3).

Examples

```
D1 <- as.matrix(dist(matrix(rnorm(20), 5)))
D2 <- as.matrix(dist(matrix(rnorm(20), 5)))
morie_spatial_voting_indscal(list(D1, D2), n_dims = 2L, max_iter = 30L)
```

`morie_spatial_voting_mds_fit_stats` *MDS fit statistics (Mardia criterion)*

Description

MDS fit statistics (Mardia criterion)

Usage

```
morie_spatial_voting_mds_fit_stats(eigenvalues)
```

Arguments

`eigenvalues`	Numeric vector of MDS eigenvalues.

Value

List with `fit_by_dim`, `cumulative_fit`, `eigenvalues`.

References

Mardia, K. V. (1978). "Some Properties of Classical Multi-Dimensional Scaling." *Communications in Statistics*.

Examples

```
morie_spatial_voting_mds_fit_stats(c(4, 2, 1, 0.5))
```

`morie_spatial_voting_mlsmu6` *MLSMU6 alternating least-squares unfolding*

Description

Multidimensional Least-Squares Metric Unfolding (Poole 1984; Bakker & Poole 2013). Alternates between respondent and stimulus coordinates, restarting from random seeds and keeping the lowest-stress fit.

Usage

```
morie_spatial_voting_mlsmu6(
  D,
  n_dims = 2L,
  max_iter = 200L,
  tol = 1e-06,
  n_restarts = 5L
)
```

Arguments

`D`	Respondent-by-stimulus distance/rating matrix.
`n_dims`	Number of latent dimensions.
`max_iter`	Maximum alternations per restart.
`tol`	Convergence tolerance on relative stress change.
`n_restarts`	Number of random restarts.

Value

A list with `respondent_coords`, `stimulus_coords`, `stress`, `iterations`, `converged`.

References

Poole, K. T. (1984). "Least Squares Metric, Unidimensional Unfolding." *Psychometrika*, 49(3). Bakker, R. and Poole, K. T. (2013).

Examples

```
D <- matrix(stats::runif(20 * 6), 20, 6)
morie_spatial_voting_mlsmu6(D, n_dims = 2, n_restarts = 1, max_iter = 50)
```

`morie_spatial_voting_nominate_bootstrap` *Parametric bootstrap of NOMINATE standard errors*

Description

Lewis & Poole (2004) parametric bootstrap: simulate roll-call matrices from fitted probabilities, re-estimate per bootstrap replicate, compute SE from the bootstrap distribution.

Usage

```
morie_spatial_voting_nominate_bootstrap(
  votes,
  ideal_points,
  normal_vectors_arr,
  cutpoints,
  n_boot = 100L,
  seed = 42L
)
```

Arguments

`votes`	Original vote matrix.
`ideal_points`	Fitted ideal points.
`normal_vectors_arr`	Fitted normal vectors.
`cutpoints`	Fitted cutpoints.
`n_boot`	Number of bootstrap replications.
`seed`	RNG seed.

Value

List with `se_ideal_points`, `boot_means`, `n_boot`.

References

Lewis, J. B. and Poole, K. T. (2004). "Measuring Bias and Uncertainty in Ideal Point Estimates via the Parametric Bootstrap." *Political Analysis*, 12(2).

Examples

```
set.seed(1)
v <- matrix(stats::rbinom(40, 1, 0.5), 5, 8)
fit <- morie_spatial_voting_dw_nominate(v, max_iter = 5L)
morie_spatial_voting_nominate_bootstrap(
  v, fit$ideal_points, fit$normal_vectors, fit$cutpoints, n_boot = 5L)
```

`morie_spatial_voting_nominate_loglik` *NOMINATE log-likelihood and GMP*

Description

NOMINATE log-likelihood and GMP

Usage

```
morie_spatial_voting_nominate_loglik(
  votes,
  x,
  z_yea,
  z_nay,
  beta = 15,
  w = NULL
)
```

Arguments

`votes`	Legislator-by-vote binary matrix.
`x`	Ideal points.
`z_yea`	Yea outcomes.
`z_nay`	Nay outcomes.
`beta`	Signal-to-noise.
`w`	Dimension weights.

Value

List with `loglik`, `GMP`, `n_correct`, `n_total`.

References

Poole & Rosenthal (1997).

Examples

```
v <- matrix(stats::rbinom(20, 1, 0.5), 4, 5)
x <- matrix(rnorm(4), 4, 1); zy <- matrix(rnorm(5), 5, 1)
zn <- matrix(rnorm(5), 5, 1)
morie_spatial_voting_nominate_loglik(v, x, zy, zn)
```

morie_spatial_voting_nominate_utility

NOMINATE Gaussian utility

Description

Computes Poole-Rosenthal NOMINATE utilities and vote probabilities.

Usage

```
morie_spatial_voting_nominate_utility(x, z_yea, z_nay, beta = 15, w = NULL)
```

Arguments

x	Legislator ideal points (n_leg x n_dims).
z_yea	Yea outcome locations (n_votes x n_dims).
z_nay	Nay outcome locations (n_votes x n_dims).
beta	Signal-to-noise ratio.
w	Dimension weights (length n_dims; defaults to 1).

Value

List with `U_yea`, `U_nay`, `utility_diff`, `vote_probs`.

References

Poole, K. T. and Rosenthal, H. (1985); Armstrong et al. (2021), Ch. 5.

Examples

```
x  <- matrix(rnorm(8), 4, 2)
zy <- matrix(rnorm(6), 3, 2); zn <- matrix(rnorm(6), 3, 2)
morie_spatial_voting_nominate_utility(x, zy, zn)
```

morie_spatial_voting_nominate_vote_prob

Single NOMINATE vote probability

Description

Single NOMINATE vote probability

Usage

```
morie_spatial_voting_nominate_vote_prob(
  x_i,
  z_yea_j,
  z_nay_j,
  beta = 15,
  w = NULL
)
```

Arguments

x_i	Legislator ideal point (vector).
z_yea_j	Yea outcome (vector).
z_nay_j	Nay outcome (vector).
beta	Signal-to-noise ratio.
w	Dimension weights.

Value

Numeric scalar in (0,1).

References

Poole & Rosenthal (1985).

Examples

```
morie_spatial_voting_nominate_vote_prob(c(0.1), c(0.3), c(-0.3))
```

---

morie_spatial_voting_nonmetric_mds

Nonmetric MDS with isotonic regression

Description

Kruskal-style nonmetric MDS using pool-adjacent-violators monotone regression on dissimilarity ranks.

Usage

```
morie_spatial_voting_nonmetric_mds(
  D,
  n_dims = 2L,
  max_iter = 300L,
  tol = 1e-06
)
```

Arguments

`D`	Symmetric dissimilarity matrix.
`n_dims`	Number of dimensions.
`max_iter`	Maximum iterations.
`tol`	Convergence tolerance.

Value

A list with `coordinates`, `stress`, `iterations`, `converged`.

References

Kruskal, J. B. (1964). "Nonmetric Multidimensional Scaling: A Numerical Method." *Psychometrika*, 29(2), 115-129.

Examples

```
D <- as.matrix(dist(matrix(rnorm(40), 10)))
morie_spatial_voting_nonmetric_mds(D)
```

`morie_spatial_voting_nonparametric_bootstrap` *Nonparametric bootstrap for AM / blackbox scaling positions*

Description

Efron & Tibshirani (1993) resampling of respondents for Aldrich-McKelvey and Basic Space scaling SEs.

Usage

```
morie_spatial_voting_nonparametric_bootstrap(
  Z,
  scale_fn = "am",
  n_boot = 200L,
  seed = 42L
)
```

Arguments

`Z`	Perception matrix.
`scale_fn`	One of `"am"`, `"blackbox"`, `"blackbox_t"`.
`n_boot`	Number of bootstrap replications.
`seed`	RNG seed.

Value

List with `se_positions`, `boot_mean`, `ci_lower`, `ci_upper`, `n_boot`.

References

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Examples

```
set.seed(1)
Z <- matrix(rnorm(20 * 5), 20, 5)
morie_spatial_voting_nonparametric_bootstrap(Z, n_boot = 10L)
```

`morie_spatial_voting_normal_vectors` *Normal-vector projection of an external measure*

Description

Normal-vector projection of an external measure

Usage

```
morie_spatial_voting_normal_vectors(ideal_points, external_measure)
```

Arguments

`ideal_points`	Ideal-point coordinates.
`external_measure`	Vector to project.

Value

List with `normal_vector`, `angle_degrees`, `angle_radians`, `r_squared`, `coefficients`.

References

Armstrong et al. (2021), Section 2.6.

Examples

```
morie_spatial_voting_normal_vectors(matrix(rnorm(20), 10, 2), rnorm(10))
```

`morie_spatial_voting_optimal_classification` *Optimal Classification scaling*

Description

Nonparametric ideal-point estimation from binary roll-call votes that minimises the number of classification errors (Poole 2000).

Usage

```
morie_spatial_voting_optimal_classification(
  votes,
  n_dims = 1L,
  max_iter = 500L,
  n_restarts = 10L,
  seed = 42L
)
```

Arguments

`votes`	Legislator-by-vote matrix; `1`=yea, `0`=nay, `NA`=missing.
`n_dims`	Number of latent dimensions.
`max_iter`	Maximum iterations per restart.
`n_restarts`	Random restarts (best PRE retained).
`seed`	RNG seed.

Value

A list with `ideal_points`, `cutting_normals`, `PRE`, `APRE`, `total_errors`, `null_errors`, `n_dims`.

References

Poole, K. T. (2000). "Non-Parametric Unfolding of Binary Choice Data." *Political Analysis*, 8(3), 211-237.

Examples

```
set.seed(1)
v <- matrix(stats::rbinom(20 * 30, 1, 0.5), 20, 30)
morie_spatial_voting_optimal_classification(v)
```

morie_spatial_voting_ordered_oc

Ordered Optimal Classification for ordinal scales

Description

Ordered Optimal Classification for ordinal scales

Usage

```
morie_spatial_voting_ordered_oc(Y, n_dims = 2L, max_iter = 500L, tol = 1e-06)
```

Arguments

`Y`	Respondent-by-item ordinal response matrix.
`n_dims`	Latent dimensions.
`max_iter`	Maximum iterations.
`tol`	Tolerance (unused; kept for API parity).

Value

List with `ideal_points`, `cutpoints`, `normals`, `correct_class`, `iterations`.

References

Hare, C., Liu, T.-P., and Lupton, R. N. (2018). "What Ordered Optimal Classification reveals about ideological structure, cleavages, and polarization in the American mass public." *Public Choice*, 176(1), 57-78.

Examples

```
Y <- matrix(sample(1:4, 60, replace = TRUE), 15, 4)
morie_spatial_voting_ordered_oc(Y, n_dims = 1L, max_iter = 20L)
```

morie_spatial_voting_ordinal_irt

Ordinal IRT / Quinn factor model (stub)

Description

Ordinal IRT / Quinn factor model (stub)

Usage

```
morie_spatial_voting_ordinal_irt(
  Y,
  n_dims = 1L,
  n_samples = 500L,
  burn_in = 100L,
  seed = 42L
)
```

Arguments

`Y`	Ordinal response matrix.
`n_dims`	Latent dimensions.
`n_samples`	MCMC samples. @param burn_in Burn-in length.
`burn_in`	Integer; MCMC burn-in iterations.
`seed`	RNG seed.

Value

Never returns; raises `NotYetPorted`.

References

Quinn, K. M. (2004). "Bayesian Factor Analysis for Mixed Ordinal and Continuous Responses." *Political Analysis*, 12(4).

Examples

```
## Not run: morie_spatial_voting_ordinal_irt(matrix(1L, 5, 3))
```

`morie_spatial_voting_procrustes` *Procrustes rotation*

Description

Orthogonal rotation aligning `X` to `X_target` (with reflection protection against improper rotations).

Usage

```
morie_spatial_voting_procrustes(X, X_target)
```

Arguments

`X`	Configuration to rotate.
`X_target`	Target configuration.

Value

List with `rotated`, `rotation_matrix`, `scale`, `mse`.

References

Gower & Dijksterhuis (2004).

Examples

```
A <- matrix(rnorm(20), 10, 2); B <- A + 0.05 * matrix(rnorm(20), 10, 2)
morie_spatial_voting_procrustes(A, B)
```

`morie_spatial_voting_smacof` *SMACOF stress minimisation*

Description

Iterative majorisation algorithm for metric MDS.

Usage

```
morie_spatial_voting_smacof(
  D,
  n_dims = 2L,
  max_iter = 300L,
  tol = 1e-06,
  weights = NULL,
  init = NULL
)
```

Arguments

`D`	Symmetric dissimilarity matrix.
`n_dims`	Number of dimensions.
`max_iter`	Maximum iterations.
`tol`	Convergence tolerance on stress change.
`weights`	Optional weight matrix (defaults to uniform).
`init`	Optional initial configuration (n x n_dims).

Value

A list with `coordinates`, `stress`, `iterations`, `converged`.

References

De Leeuw, J. (1977). "Applications of Convex Analysis to Multidimensional Scaling." In *Recent Developments in Statistics*, 133-145. Borg & Groenen (2005).

Examples

```
D <- as.matrix(dist(matrix(rnorm(40), 10)))
morie_spatial_voting_smacof(D)
```

---

morie_spatial_voting_smacof_unfolding

SMACOF rectangular unfolding

Description

Majorisation-based unfolding embedding respondents and stimuli into a common space.

Usage

```
morie_spatial_voting_smacof_unfolding(
  D,
  n_dims = 2L,
  max_iter = 300L,
  tol = 1e-06
)
```

Arguments

`D`	Respondent-by-stimulus dissimilarity matrix.
`n_dims`	Latent dimensions.
`max_iter`	Maximum iterations.
`tol`	Convergence tolerance.

Value

A list with `respondent_coords`, `stimulus_coords`, `stress`, `iterations`, `converged`.

References

Borg & Groenen (2005); Armstrong et al. (2021), Ch. 4.

Examples

```
D <- matrix(stats::runif(12), 3, 4)
morie_spatial_voting_smacof_unfolding(D, max_iter = 20)
```

morie_spatial_voting_unfolding_stress

Compute unfolding stress

Description

Compute unfolding stress

Usage

```
morie_spatial_voting_unfolding_stress(X_r, X_s, D, weights = NULL)
```

Arguments

X_r	Respondent coordinates (n_r x n_dims).
X_s	Stimulus coordinates (n_s x n_dims).
D	Observed respondent-stimulus dissimilarities.
weights	Optional weight matrix.

Value

A numeric scalar, the weighted sum of squared residuals.

References

Coombs (1964); Armstrong et al. (2021).

Examples

```
Xr <- matrix(rnorm(6), 3, 2); Xs <- matrix(rnorm(8), 4, 2)
D  <- matrix(stats::runif(12), 3, 4)
morie_spatial_voting_unfolding_stress(Xr, Xs, D)
```

morie_spatial_voting_wordfish

Wordfish: Poisson IRT for document-term matrices

Description

Slapin & Proksch (2008) one-dimensional Poisson IRT for estimating document positions from word-count data.

Usage

```
morie_spatial_voting_wordfish(dtm, max_iter = 100L, tol = 1e-06)
```

Arguments

`dtm`	Document-by-term integer count matrix.
`max_iter`	Maximum EM iterations.
`tol`	Convergence tolerance.

Value

List with `positions`, `word_weights`, `word_fixed`, `doc_fixed`, `log_lik`, `iterations`.

References

Slapin, J. B. and Proksch, S.-O. (2008). "A Scaling Model for Estimating Time-Series Party Positions from Texts." *AJPS*, 52(3).

Examples

```
set.seed(1)
dtm <- matrix(stats::rpois(20 * 30, 5), 20, 30)
morie_spatial_voting_wordfish(dtm, max_iter = 20L)
```

`morie_spearman_rho` *Spearman rank correlation*

Description

Spearman rank correlation

Usage

```
morie_spearman_rho(x, y)
```

Arguments

`x`	Numeric vector.
`y`	Numeric vector.

Value

Named list: `rho`, `p_value`.

Examples

```
morie_spearman_rho(x = rnorm(50), y = rnorm(50))
```

morie_specs_from_df *Build a list of column specs from a parsed CSV header.*

Description

Convenience helper: given a data.frame just-loaded by `morie_arsau_load_*()`, returns the `list(name, dtype, valid_values)` structure expected by the audit functions.

Usage

```
morie_specs_from_df(df)
```

Arguments

`df`	Loaded data.frame.

Value

List of column specs.

morie_spectral_cluster *Spectral clustering via* **kernlab**

Description

Thin extender over `kernlab::specc` that performs Ng / Jordan / Weiss spectral clustering on a feature matrix.

Usage

```
morie_spectral_cluster(x, centers, ...)
```

Arguments

`x`	Numeric matrix or data frame of features (rows = observations).
`centers`	Integer; the number of clusters to extract.
`...`	Further arguments forwarded to `kernlab::specc` (e.g. `kernel`, `kpar`, `nystrom.red`, `iterations`).

Value

A list with `$method = "kernlab::specc"` and `$raw` (a `specc` S4 object containing the cluster assignments, centres and within-cluster sums of squares).

Examples

```
## Not run:
  if (requireNamespace("kernlab", quietly = TRUE)) {
    set.seed(1)
    x <- rbind(
      matrix(stats::rnorm(80, mean = -2), ncol = 2),
      matrix(stats::rnorm(80, mean =  2), ncol = 2)
    )
    morie_spectral_cluster(x, centers = 2)
  }

## End(Not run)
```

---

morie_spectral_density

Welch power spectral density

Description

Welch power spectral density

Usage

```
morie_spectral_density(x, fs = 1, nperseg = NULL)
```

Arguments

`x`	Numeric univariate series.
`fs`	Sampling frequency. Default 1.
`nperseg`	Segment length. Default max(n/4, 8).

Value

Named list with `frequencies`, `psd`, `n_segments`, `nperseg`, `fs`, `n`, `method`.

Examples

```
morie_spectral_density(x = rnorm(50))
```

`morie_sprott_doob` *Sprott & Doob (CRIMSL UToronto) SIU analyses*

Description

Replicates the analytical contribution of the four CRIMSL UToronto research reports authored by **Prof. Jane B. Sprott** (TMU, formerly Ryerson) and **Prof. Anthony N. Doob** (University of Toronto), with **Prof. Adelina Iftene** (Dalhousie) co- authoring the May 2021 paper on Independent External Decision Makers (IEDMs).

Details

Sprott & Doob (Oct 2020) *Understanding the Operation of CSC's Structured Intervention Units* – first systematic outside analysis of CSC SIU data.

Sprott & Doob (Nov 2020) *COVID attribution* – tests CSC's COVID-attribution defense.

Sprott & Doob (Feb 2021) *Solitary Confinement, Torture, and Canada's SIUs* – introduces the Mandela-Rules classifier; the most data-intensive of the four.

Sprott, Doob & Iftene (May 2021) *Independent External Decision Makers* – evaluates the IEDM review mechanism.

Headline tables (Feb 2021): Tables 13, 19, 23 reproduce SIU person-stay rates per 1000 prisoners, the Mandela-Rules classification of N=1960 SIU stays (solitary 28.4%, torture 9.9%, all-other 61.7%), and the regional torture/solitary rates.

Headline tables (May 2021): Tables 1, 3, 5, 7, 8, 9, 10, 14, 15 reproduce IEDM-reviewed population characteristics and review outcomes (N=265 stays, 380 reviews).

Citation

Sprott, J. B., & Doob, A. N. (2021, February). Solitary Confinement, Torture, and Canada's Structured Intervention Units. Centre for Criminology & Sociolegal Studies, U. of Toronto.

`morie_stat_bridge` *Bridge between external runners and the morie R command registry*

Description

R port of `morie.stat_bridge`. Exposes the same three modes available on the Python side — registry enumeration, a formatted help dump, and command execution — so an external runner (e.g. the Go TIDE TUI, a shell pipeline) can drive morie's R surface via `Rscript -e 'rmorie::stat_bridge_main(...)'`.

Details

Two layers are provided:

1. Programmatic helpers (`stat_bridge_registry_json`, `stat_bridge_help`, `stat_bridge_exec`) callable from ordinary R code.
2. A dispatcher (`stat_bridge_main`) that mimics the command-line entry point of the Python module so the same invocation pattern works from either runtime.

`morie_stat_commands` *Central command registry for the morie R surface*

Description

R port of `morie.stat_commands`. Maintains a flat registry of R-callable statistical command entries plus aliases, allowing downstream tooling (the Go TIDE bridge, headless workers, REPL frontends) to enumerate, resolve, and dispatch operations from a single namespace.

Details

Entries are stored in the package-level environment `.morie_stat_commands` so that the registry is shared across sessions within a single R process and can be appended by extension packages.

Functions:

- `stat_command`: constructor for a single command.
- `register_stat_command`: add an entry to the registry, validating uniqueness of name and aliases.
- `resolve_stat_command`: look up by canonical name or alias.
- `all_stat_command_names`: sorted vector of all names + aliases.
- `commands_by_category`: list of entries grouped by category.
- `run_stat_command`: invoke a command's REPL handler with the supplied arguments.

`morie_state_space_model` *Local-level state-space model (Kalman filter+smoother)*

Description

Local-level state-space model (Kalman filter+smoother)

Usage

```
morie_state_space_model(x)
```

Arguments

x	Numeric univariate series.

Value

Named list with filtered_state, filtered_state_variance, smoothed_state, loglik, Q, R, n, method.

Examples

```
morie_state_space_model(x = rnorm(50))
```

---

morie_stratified_sample

Proportional or fixed stratified random sample

Description

Proportional or fixed stratified random sample

Usage

```
morie_stratified_sample(
  df,
  strata_col,
  n_per_stratum,
  proportional = FALSE,
  seed = 42L
)
```

Arguments

df	A data frame.
strata_col	Name of the stratification column.
n_per_stratum	Either an integer (equal allocation) or a named integer vector mapping stratum levels to sample sizes. If proportional = TRUE, n_per_stratum is treated as the total desired sample size and allocation is proportional to stratum size.
proportional	Logical; if TRUE, allocate proportionally to strata sizes.
seed	Random seed.

Value

Data frame of sampled rows with a .weight column.

Examples

```
df <- data.frame(g = c(rep("A", 60), rep("B", 40)), x = rnorm(100))
morie_stratified_sample(df, "g", n_per_stratum = 10)
```

morie_suggest_analysis_plan

Suggest an analysis plan from a dataset profile

Description

Mirrors the Python `morie.suggest_analysis_plan()`. Inspects the output of `morie_profile_dataset()` and returns plain-English recommendations for candidate analyses.

Usage

```
morie_suggest_analysis_plan(profile)
```

Arguments

`profile`	A list returned by `morie_profile_dataset()`.

Value

Character vector of suggestion strings, one per recommendation.

Examples

```
morie_suggest_analysis_plan(morie_profile_dataset(iris))
```

morie_sukhatme_test

Sukhatme two-sample scale test (Gibbons Ch 9.7)

Description

Mann-Whitney U on the absolute deviations from the pooled median. Tests equality of scales given (approximately) equal medians.

Usage

```
morie_sukhatme_test(x, y)
```

Arguments

`x`, `y`	Numeric vectors.

Value

Named list: statistic (z), p_value, U, n, m.

Examples

```
morie_sukhatme_test(x = rnorm(50), y = rnorm(50))
```

`morie_summarize_output_audit`
Summarize an output audit

Description

Summarize an output audit

Usage

```
morie_summarize_output_audit(audit_tbl)
```

Arguments

`audit_tbl`	Result from `morie_audit_public_outputs()`.

Value

Named list with high-level diagnostics.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_survey_calibrate`
Raking calibration to known marginal totals (iterative proportional fitting).

Description

For multi-variable marginals use `morie_weights_rake()`; this helper is the single-variable convenience.

Usage

```
morie_survey_calibrate(
  df,
  aux_vars,
  population_totals,
  max_iter = 50,
  tol = 1e-06
)
```

Arguments

`df`	A `data.frame` holding the (unit-level) sample data plus any weight/strata/cluster/domain columns referenced by name.
`aux_vars`	Character vector of column names of auxiliary variables used for calibration (raking, GREG, etc.).
`population_totals`	Named numeric vector of population totals to calibrate to (one entry per `aux_vars` element).
`max_iter`	Iteration cap for the iterative calibration loop.
`tol`	Convergence tolerance for calibration.

`morie_survey_complex_glm`

Complex-survey GLM constructor (single-shot wrapper that builds a design and fits a `svyglm` *in one call). Cluster-robust SEs via the design.*

Description

Complex-survey GLM constructor (single-shot wrapper that builds a design and fits a `svyglm` in one call). Cluster-robust SEs via the design.

Usage

```
morie_survey_complex_glm(
  df,
  formula,
  weight_col,
  family = "gaussian",
  cluster_col = NULL,
  strata_col = NULL
)
```

Arguments

`df`	A `data.frame` holding the (unit-level) sample data plus any weight/strata/cluster/domain columns referenced by name.
`formula`	A `formula` (e.g. `y ~ x1 + x2`) for survey-weighted regression / GLM.
`weight_col`	Character; column name of the design weight variable in `df`.
`family`	A `family` object (e.g. `stats::binomial()`) passed to the survey-weighted GLM.
`cluster_col`	Character; column name of the cluster identifier in `df`.
`strata_col`	Character; column name of the stratum identifier in `df`.

morie_survey_design *Construct a survey design object.*

Description

Returns a `survey::svydesign` object when `survey` is available; otherwise returns a lightweight list with the same fields the morie helpers consume.

Usage

```
morie_survey_design(
  data,
  weights_col,
  strata_col = NULL,
  cluster_col = NULL,
  fpc_col = NULL,
  nest = FALSE
)
```

Arguments

`data`	data.frame.
`weights_col`	Column name of analytic/probability weights.
`strata_col`	Optional strata column.
`cluster_col`	Optional PSU/cluster column.
`fpc_col`	Optional finite-population-correction column.
`nest`	If TRUE, treat cluster IDs as nested within strata.

morie_survey_glm *Survey-weighted GLM with design-based SEs.*

Description

Wraps `survey::svyglm()`. Family argument accepts the same strings as the Python module ("gaussian", "binomial", "poisson", "gamma", "negativebinomial") or any R `family` object.

Usage

```
morie_survey_glm(
  design,
  formula,
  family = c("gaussian", "binomial", "poisson", "gamma", "negativebinomial")
)
```

Arguments

`design`	A `survey::svydesign` object (when interfacing with the `survey` package directly).
`formula`	A `formula` (e.g. `y ~ x1 + x2`) for survey-weighted regression / GLM.
`family`	A `family` object (e.g. `stats::binomial()`) passed to the survey-weighted GLM.

`morie_survey_hajek_mean` *Hajek (ratio) estimator of a population mean.*

Description

Hajek (ratio) estimator of a population mean.

Usage

```
morie_survey_hajek_mean(y, weights)
```

Arguments

`y`	Numeric vector of outcome values aligned with the sample.
`weights`	Numeric vector of design weights aligned with `y`.

`morie_survey_ht_total` *Horvitz-Thompson estimator of a population total.*

Description

Horvitz-Thompson estimator of a population total.

Usage

```
morie_survey_ht_total(y, inclusion_probs)
```

Arguments

`y`	Numeric vector of outcome values aligned with the sample.
`inclusion_probs`	Numeric vector of inclusion probabilities (`pi_i`) for the Horvitz-Thompson estimator.

Value

list with `total`, `se`, `ci_lower`, `ci_upper`.

`morie_survey_mean` *Survey-weighted mean (delegates to* `survey::svymean` *when available).*

Description

Survey-weighted mean (delegates to `survey::svymean` when available).

Usage

```
morie_survey_mean(design, variable)
```

Arguments

`design`	A `survey::svydesign` object (when interfacing with the `survey` package directly).
`variable`	Character; column name of the outcome variable in the `design` object (`morie_survey_mean`).

`morie_survey_poststratify`
Post-stratification weights (sample-to-population alignment).

Description

Delegates to `survey::postStratify()` when given a design; otherwise computes raw post-stratification factors in base R.

Usage

```
morie_survey_poststratify(df, strata_col, population_counts)
```

Arguments

`df`	A `data.frame` holding the (unit-level) sample data plus any weight/strata/cluster/domain columns referenced by name.
`strata_col`	Character; column name of the stratum identifier in `df`.
`population_counts`	Named numeric vector of population counts by stratum (post-stratification target).

`morie_survey_ratio` *Ratio estimator of a population total using known X_pop.*

Description

Ratio estimator of a population total using known X_pop.

Usage

```
morie_survey_ratio(y, x, weights, X_population_total)
```

Arguments

`y`	Numeric vector of outcome values aligned with the sample.
`x`	Numeric vector of auxiliary values aligned with `y` (used in ratio estimation).
`weights`	Numeric vector of design weights aligned with `y`.
`X_population_total`	Known population total for the auxiliary variable `x` (ratio estimator).

`morie_survey_subpop` *Subpopulation (domain) mean with Woodruff linearised SE.*

Description

Subpopulation (domain) mean with Woodruff linearised SE.

Usage

```
morie_survey_subpop(df, domain_col, domain_value, outcome_col, weight_col)
```

Arguments

`df`	A `data.frame` holding the (unit-level) sample data plus any weight/strata/cluster/domain columns referenced by name.
`domain_col`	Character; column name of the subpopulation / domain indicator in `df`.
`domain_value`	Value (matching `df[[domain_col]]`) defining the subpopulation to estimate.
`outcome_col`	Character; column name of the outcome variable in `df`.
`weight_col`	Character; column name of the design weight variable in `df`.

morie_survival_aft *Accelerated failure time model (parametric).*

Description

Wraps `survival::survreg()`. Supported `dist`: "weibull", "lognormal", "loglogistic", "exponential", "gaussian".

Usage

```
morie_survival_aft(
  data,
  duration_col,
  event_col,
  covariate_cols,
  dist = c("weibull", "lognormal", "loglogistic", "exponential", "gaussian")
)
```

Arguments

`data`	A `data.frame` whose columns supply `duration_col`, `event_col`, `covariate_cols`, etc.
`duration_col`	Character; column name of the event/censoring time in `data`.
`event_col`	Character; column name of the event-indicator variable in `data`.
`covariate_cols`	Character vector of covariate column names.
`dist`	Distribution name for parametric/AFT fits (e.g. `"weibull"`, `"lognormal"`, `"loglogistic"`).

morie_survival_cif *Cumulative incidence function (Aalen-Johansen) for competing risks.*

Description

Wraps `survival::survfit()` with multi-state `Surv()`.

Usage

```
morie_survival_cif(time, event, event_of_interest = 1L, confidence = 0.95)
```

Arguments

`time`	Numeric vector of event/censoring times.
`event`	Integer event code: 0 = censored, 1 = event of interest,=2 = competing event.
`event_of_interest`	Integer event-type code for competing- risks analyses (Fine-Gray, CIF).
`confidence`	Confidence level for interval estimates (default `0.95`).

`morie_survival_compare_parametric`

Compare parametric survival models by AIC/BIC.

Description

Compare parametric survival models by AIC/BIC.

Usage

```
morie_survival_compare_parametric(time, event)
```

Arguments

`time`	Numeric vector of event/censoring times.
`event`	Integer/logical vector; 1 = event, 0 = censored.

`morie_survival_concordance`

Harrell's concordance index (C-statistic).

Description

Uses `survival::concordance()` (which handles ties + censoring correctly).

Usage

```
morie_survival_concordance(time, event, risk_score)
```

Arguments

`time`	Numeric vector of event/censoring times.
`event`	Integer/logical vector; 1 = event, 0 = censored.
`risk_score`	Numeric vector of predicted risk scores aligned with `time` for concordance / discrimination metrics.

morie_survival_cox *Cox proportional hazards model.*

Description

Wraps `survival::coxph()` with Efron (default) or Breslow tie handling and returns a tidy list including hazard ratios, CIs, p-values, and the Breslow baseline cumulative hazard.

Usage

```
morie_survival_cox(
  data,
  duration_col,
  event_col,
  covariate_cols,
  ties = c("efron", "breslow"),
  confidence = 0.95,
  penalizer = 0
)
```

Arguments

`data`	data.frame.
`duration_col`	Name of the time column.
`event_col`	Name of the 0/1 event column.
`covariate_cols`	Character vector of covariate column names.
`ties`	"efron" (default) or "breslow".
`confidence`	Confidence level (default 0.95).
`penalizer`	L2 penalty (passed via `ridge()` term in the formula).

morie_survival_coxsnell *Cox-Snell residuals from a fitted morie Cox model.*

Description

Cox-Snell residuals from a fitted morie Cox model.

Usage

```
morie_survival_coxsnell(cox_result)
```

Arguments

`cox_result`	A `coxph`-style fit (as returned by `morie_survival_cox`); used by residual diagnostics (`coxsnell`, `martingale`, `deviance`).

`morie_survival_deviance`
Deviance residuals.

Description

Deviance residuals.

Usage

```
morie_survival_deviance(cox_result)
```

Arguments

`cox_result`	A `coxph`-style fit (as returned by `morie_survival_cox`); used by residual diagnostics (`coxsnell`, `martingale`, `deviance`).

`morie_survival_finegray`
Fine-Gray subdistribution hazard model (competing risks).

Description

Requires the `cmprsk` package.

Usage

```
morie_survival_finegray(
  data,
  duration_col,
  event_col,
  covariate_cols,
  event_of_interest = 1L,
  confidence = 0.95
)
```

Arguments

`data`	A `data.frame` whose columns supply `duration_col`, `event_col`, `covariate_cols`, etc.
`duration_col`	Character; column name of the event/censoring time in `data`.
`event_col`	Character; column name of the event-indicator variable in `data`.
`covariate_cols`	Character vector of covariate column names.
`event_of_interest`	Integer event-type code for competing- risks analyses (Fine-Gray, CIF).
`confidence`	Confidence level for interval estimates (default `0.95`).

morie_survival_hr *Hazard ratio between two groups via a simple Cox model.*

Description

Hazard ratio between two groups via a simple Cox model.

Usage

```
morie_survival_hr(time, event, group, confidence = 0.95)
```

Arguments

time	Numeric vector of event/censoring times.
event	Integer/logical vector; 1 = event, 0 = censored.
group	Factor/character grouping variable for HR / log-rank stratified comparisons.
confidence	Confidence level for interval estimates (default 0.95).

morie_survival_km *Kaplan-Meier product-limit survival estimator.*

Description

Thin wrapper around survival::survfit() returning a tidy list with Greenwood or complementary-log-log confidence bands.

Usage

```
morie_survival_km(
  time,
  event,
  confidence = 0.95,
  ci_method = c("greenwood", "log-log")
)
```

Arguments

time	Numeric vector of observation times.
event	0/1 event indicator (1 = event observed).
confidence	Confidence level (default 0.95).
ci_method	"greenwood" (plain) or "log-log".

Value

list with times, survival, ci_lower, ci_upper, at_risk, events, censored, median_survival, method.

`morie_survival_landmark`
Landmark dataset constructor.

Description

Landmark dataset constructor.

Usage

```
morie_survival_landmark(data, duration_col, event_col, landmark_time)
```

Arguments

`data`	A `data.frame` whose columns supply `duration_col`, `event_col`, `covariate_cols`, etc.
`duration_col`	Character; column name of the event/censoring time in `data`.
`event_col`	Character; column name of the event-indicator variable in `data`.
`landmark_time`	Numeric landmark time at which to subset the cohort before fitting (lead-time bias correction).

`morie_survival_left_truncated_km`
Left-truncated Kaplan-Meier with delayed entry.

Description

Left-truncated Kaplan-Meier with delayed entry.

Usage

```
morie_survival_left_truncated_km(
  entry_time,
  exit_time,
  event,
  confidence = 0.95
)
```

Arguments

`entry_time`	Left-truncation entry times.
`exit_time`	Exit (event/censoring) times for the left-truncated KM estimator.
`event`	Integer/logical vector; 1 = event, 0 = censored.
`confidence`	Confidence level for interval estimates (default `0.95`).

`morie_survival_logrank`

Log-rank family tests (logrank / Peto-Peto / Gehan / Tarone-Ware).

Description

Delegates to `survival::survdiff()` for the standard log-rank weight (rho=0) and Peto-Peto (rho=1). Gehan/Tarone-Ware are not supported by `survdiff` directly and currently fall back to rho=1 (Peto) as the closest analogue; use `survival::survdiff(..., rho=1)` plus FH weights for exact equivalents.

Usage

```
morie_survival_logrank(
  time,
  event,
  group,
  weight = c("logrank", "peto", "gehan", "tarone")
)
```

Arguments

`time`, `event`, `group`	Vectors.
`weight`	One of "logrank", "peto", "gehan", "tarone".

`morie_survival_martingale`

Martingale residuals.

Description

Martingale residuals.

Usage

```
morie_survival_martingale(cox_result)
```

Arguments

`cox_result`	A `coxph`-style fit (as returned by `morie_survival_cox`); used by residual diagnostics (`coxsnell`, `martingale`, `deviance`).

`morie_survival_nelsonaalen`
Nelson-Aalen cumulative-hazard estimator.

Description

Nelson-Aalen cumulative-hazard estimator.

Usage

```
morie_survival_nelsonaalen(time, event, confidence = 0.95)
```

Arguments

`time`	Numeric vector of observation times.
`event`	0/1 event indicator (1 = event observed).
`confidence`	Confidence level (default 0.95).

Value

list with `times`, `cumhaz`, `ci_lower`, `ci_upper`, `at_risk`, `events`, `censored`.

`morie_survival_parametric`
Simple parametric survival models (intercept-only).

Description

For "exponential", "weibull", "lognormal", "loglogistic", "gaussian". Use `morie_survival_aft()` for covariate-adjusted parametric models.

Usage

```
morie_survival_parametric(
  time,
  event,
  dist = c("weibull", "exponential", "lognormal", "loglogistic", "gaussian")
)
```

Arguments

`time`	Numeric vector of event/censoring times.
`event`	Integer/logical vector; 1 = event, 0 = censored.
`dist`	Distribution name for parametric/AFT fits (e.g. `"weibull"`, `"lognormal"`, `"loglogistic"`).

morie_survival_rmst *Restricted Mean Survival Time (RMST).*

Description

Integrates the Kaplan-Meier estimator from 0 to `tau` using trapezoidal integration on the step-function. SE follows the Klein-Moeschberger formula (approximation matches the Python module).

Usage

```
morie_survival_rmst(time, event, tau = NULL, confidence = 0.95)
```

Arguments

`time`	Numeric vector of event/censoring times.
`event`	Integer/logical vector; 1 = event, 0 = censored.
`tau`	RMST truncation horizon (`morie_survival_rmst/rmst_diff`).
`confidence`	Confidence level for interval estimates (default `0.95`).

morie_survival_rmst_diff *Difference in RMST between two groups.*

Description

Difference in RMST between two groups.

Usage

```
morie_survival_rmst_diff(
  time1,
  event1,
  time2,
  event2,
  tau = NULL,
  confidence = 0.95
)
```

Arguments

`time1`	Time vector for group 1 (`rmst_diff`).
`event1`	Event vector for group 1 (`rmst_diff`).
`time2`	Time vector for group 2 (`rmst_diff`).
`event2`	Event vector for group 2 (`rmst_diff`).
`tau`	RMST truncation horizon (`morie_survival_rmst/rmst_diff`).
`confidence`	Confidence level for interval estimates (default `0.95`).

`morie_survival_schoenfeld`

Schoenfeld residuals + PH-assumption test.

Description

Wraps `survival::cox.zph()` (scaled Schoenfeld residuals).

Usage

```
morie_survival_schoenfeld(cox_result)
```

Arguments

`cox_result`	Object returned by `morie_survival_cox()`.

`morie_survival_turnbull`

Turnbull NPMLE for interval-censored data.

Description

Delegates to `survival::survfit()` with `Surv(left, right, type = "interval2")`. Hand-rolled EM is left as a stub for environments without `survival`.

Usage

```
morie_survival_turnbull(left, right, max_iter = 200, tol = 1e-06)
```

Arguments

`left`	Left-bracket times for interval-censored data (`morie_survival_turnbull`).
`right`	Right-bracket times for interval-censored data.
`max_iter`	Iteration cap for the Turnbull NPMLE EM loop.
`tol`	Convergence tolerance for the Turnbull EM.

morie_svm_genomic *Support-vector regression for genomic prediction*

Description

Support-vector regression for genomic prediction

Usage

```
morie_svm_genomic(x, y, markers, C = 1, epsilon = 0.1, gamma = "scale")
```

Arguments

x	Optional fixed-effect features.
y	Numeric response.
markers	(n x m) genotype matrix.
C	Cost (default 1).
epsilon	SVR tube width (default 0.1).
gamma	RBF kernel scale ("scale" = 1/(m * var(M)) or numeric).

Value

list(estimate, y_hat, alpha, support_indices, se, n, method).

References

Vapnik (1995); Montesinos Lopez Ch 7.

Examples

```
morie_svm_genomic(x = rnorm(50), y = rnorm(50), markers = matrix(sample(0:2, 200, TRUE), 50, 4))
```

morie_svm_hinge_primal *Linear SVM (primal hinge loss) – R parity*

Description

Wraps `e1071::svm` with a linear kernel.

Usage

```
morie_svm_hinge_primal(x, y, C = 1, seed = 0L)
```

Arguments

x	Numeric predictor matrix.
y	Binary response.
C	Soft-margin inverse regularisation.
seed	RNG seed.

Value

Named list: estimate, intercept, weights, train_accuracy, C, classes, n, method.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

---

morie_svm_kernel_trick

Kernel SVM (RBF / poly / sigmoid) – R parity

Description

Wraps `e1071::svm`.

Usage

```
morie_svm_kernel_trick(
  x,
  y,
  kernel = "rbf",
  C = 1,
  gamma = "scale",
  degree = 3L,
  seed = 0L
)
```

Arguments

x	Numeric predictor matrix.
y	Binary response.
kernel	One of "rbf" (radial), "poly", "sigmoid", "linear".
C	Cost parameter.
gamma	Kernel coefficient ("scale" -> 1/(ncol(x)*var(x)), "auto" -> 1/p, or numeric).
degree	Polynomial degree.
seed	RNG seed.

Value

Named list: estimate, train_accuracy, n_support, kernel, C, gamma, degree, n, method.

Examples

```
morie_svm_kernel_trick(x = rnorm(50), y = rnorm(50))
```

---

morie_sync_rng *Synchronised RNG seeded reproducibly for cross-language workflows*

Description

Returns a function that mimics `stats::runif` but is seeded from `seed`. Pairs with `morie.longitudinal_sim.sync` on the Python side so the two emit identical streams when given the same seed.

Usage

```
morie_sync_rng(seed)
```

Arguments

`seed`	Non-negative integer seed.

Value

An environment with `rnorm`, `runif`, `sample` methods that share the same underlying RNG state.

Examples

```
morie_sync_rng(seed = 1L)
```

---

morie_synth_corrections_uof *Build a synthetic Corrections-UoF data.frame for testing.*

Description

Returns a small data.frame mirroring the column shape of the published corrections-UoF resource for the given short `key`. Schemas are derived from the bundled `inst/extdata/corrections_uof_dictionar` (parsed from `datadictionary_correctionsribd_en_fr20250822.xlsx`). Values are uniformly drawn from the dictionary "Data Values" examples or generic ranges; used only for offline-no-fixture fallback in tests and demos. NOT a substitute for the real upstream data published at https://data.ontario.ca/dataset/use-of-force-in-correctional-institutions.

Usage

```
morie_synth_corrections_uof(key, n = 30L, seed = 1L)
```

Arguments

key	Character; one of the 12 short names returned by morie_corrections_uof_resource_ids.
n	Integer; number of synthetic rows. Default 30.
seed	Integer; RNG seed. Default 1.

Value

A data.frame.

See Also

morie_datasets_corrections_uof_incidents and the 11 sibling loaders.

Examples

```
df <- morie_synth_corrections_uof("incidents", n = 10)
nrow(df)
```

---

morie_synth_otis *Build a synthetic OTIS data.frame for a given publication id.*

Description

Returns a data.frame mirroring the column shape + categorical level set of the published OTIS dataset for the given id (a01, b01..b09, c01..c12, or d01..d07). Schema is derived from the Ontario MCSCS XLSX data dictionary that ships at inst/extdata/otis_dictionary.json. Values are randomly drawn from the dictionary categorical levels (or 0..80 for count columns); used only for offline-no-fixture fallback in tests and demos. NOT a substitute for the real OTIS data published at https://data.ontario.ca/dataset/data-on-inmates-in-ontario.

Usage

```
morie_synth_otis(id, n = 200L, seed = 1L)
```

Arguments

id	Character; one of "a01", "b01".."b09", "c01".."c12", or "d01".."d07". Unknown ids fall through to the b01 person-level panel.
n	Integer; number of synthetic rows. Default 200.
seed	Integer; RNG seed for reproducibility. Default 1.

Value

A data.frame with the dictionary-derived schema.

See Also

morie_synth_otis_all for the full 29-dataset list; morie_datasets_otis_a01 and friends for the real bundled+live loaders.

Examples

```
df <- morie_synth_otis("c11", n = 50)
head(df)
```

---

morie_synth_otis_all *Build the full 29-dataset OTIS synthetic list.*

Description

Returns a named list of synthetic data.frames keyed by OTIS publication id (a01, b01..b09, c01..c12, d01..d07). Each frame is built by morie_synth_otis with a per-id seed offset for reproducibility.

Usage

```
morie_synth_otis_all(n = 80L, seed = 2L)
```

Arguments

n	Integer; rows per dataset. Default 80.
seed	Integer; base RNG seed. Default 2.

Value

Named list of 29 data.frames.

See Also

morie_synth_otis.

Examples

```
all_otis <- morie_synth_otis_all(n = 30)
names(all_otis)
```

`morie_terry_hoeffding_test` *Terry-Hoeffding (Fisher-Yates) two-sample normal-scores test (Gibbons Ch 8.3.1)*

Description

Replaces pooled ranks with Blom-approximated normal scores a_i = qnorm((R_i - 3/8) / (N + 1/4)). Statistic stat_t = sum of scores from the first sample.

Usage

```
morie_terry_hoeffding_test(x, y)
```

Arguments

`x`, `y`	Numeric vectors.

Value

Named list: statistic, p_value, z, n, m.

Examples

```
morie_terry_hoeffding_test(x = rnorm(50), y = rnorm(50))
```

`morie_tgarch_model` *GJR-GARCH(1,1) threshold GARCH*

Description

GJR-GARCH(1,1) threshold GARCH

Usage

```
morie_tgarch_model(x)
```

Arguments

`x`	Numeric return series.

Value

Named list with `omega`, `alpha`, `gamma`, `beta`, `persistence`, `loglik`, `conditional_variance`, `n`, `method`.

Examples

```
morie_tgarch_model(x = rnorm(50))
```

`morie_threshold_autoregression` *Two-regime self-exciting threshold autoregressive (SETAR) model*

Description

Two-regime self-exciting threshold autoregressive (SETAR) model

Usage

```
morie_threshold_autoregression(x, p = 1, d = 1, n_grid = 50)
```

Arguments

`x`	Numeric univariate series.
`p`	AR order in each regime. Default 1.
`d`	Delay parameter for the threshold variable. Default 1.
`n_grid`	Grid size for threshold search. Default 50.

Value

Named list with `threshold`, `phi_lower`, `phi_upper`, `p`, `d`, `regime_sizes`, `sse`, `n`, `method`.

Examples

```
morie_threshold_autoregression(x = rnorm(50))
```

`morie_to_hood_crosswalk` *Load the bundled BIDIRECTIONAL 158 <-> 140 neighbourhood crosswalk*

Description

Returns the bundled `inst/extdata/to_hood_158_140_crosswalk.csv`, computed from polygon intersection of the two upstream Open Toronto GeoJSON layers (`Neighbourhoods - 4326.geojson` and `Neighbourhoods - historical 140 - 4326.geojson`) reprojected to EPSG:3347 (NAD83 Statistics Canada Lambert – metric, accurate areas). Seven columns:

Usage

```
morie_to_hood_crosswalk()
```

Details

hood_140 3-char zero-padded historical short code

name_140 historical neighbourhood name (carries "(NN)" suffix)

hood_158 3-char zero-padded current short code

name_158 current neighbourhood name

pct_140_in_158 FORWARD: percent of the 140's area inside this 158. Per-140 rows sum to 100. Used by `morie_tps_disaggregate_140_to_158()` as the cake-cutting weight under a uniform-density assumption.

pct_158_in_140 REVERSE: percent of the 158's area inside this 140. Per-158 rows sum to 100. For pure cake-cuts every split child has `pct_158_in_140 == 100` (each 158 is entirely inside its parent 140), so `morie_tps_aggregate_158_to_140()` is mathematically EXACT (lossless sum) for the 1:1 + split cohort. Only the one split+merge sliver in the bundled OT data has a non-100 reverse percent.

relation "1:1" / "split" (one 140 -> N 158s) / "merge" (multiple 140s -> one 158) / "split+merge"

Empirical distribution on the bundled OT data:

- 123 1:1 rows (78\
- 34 split rows (16 historical hoods) – pct_158_in_140 == 100
- 1 merge – both percents == 100
- 1 split+merge – one sliver < 100

Value

A `data.frame` with the columns above. `hood_140` and `hood_158` are character (zero-padded to 3 chars).

`morie_to_neighbourhoods` *Load a Toronto neighbourhood polygon layer*

Description

Returns a data.frame with the canonical City of Toronto Open Data schema (`_id`, `AREA_ID`, `AREA_ATTR_ID`, `PARENT_AREA_ID`, `AREA_SHORT_CODE`, `AREA_LONG_CODE`, `AREA_NAME`, `AREA_DESC`, `CLASSIFICATION`, `CLASSIFICATION_CODE`, `OBJECTID`, `geometry`) for the requested version.

Usage

```
morie_to_neighbourhoods(
  version = c("158", "140", "nia"),
  offline = TRUE,
  resource_id = NULL
)
```

Arguments

`version`	One of `"158"` (current City scheme), `"140"` (historical 2014–2021 scheme), or `"nia"` (Neighbourhood Improvement Areas).
`offline`	If `TRUE` (default), read the small bundled synthetic fixture from `inst/extdata/`. If `FALSE`, hit the live City of Toronto CKAN `datastore_search` endpoint via httr2.
`resource_id`	Optional CKAN resource id override. Used only when `offline = FALSE`.

Value

A `data.frame`.

References

City of Toronto Open Data, "Neighbourhoods" dataset (https://open.toronto.ca/dataset/neighbourhoods/); "Neighbourhood Improvement Areas" (https://open.toronto.ca/dataset/neighbourhood-improvement-areas/); licensed under the Open Government Licence – Toronto.

Examples

```
df <- morie_to_neighbourhoods("158", offline = TRUE)
head(df[, c("AREA_SHORT_CODE", "AREA_NAME")])
```

---

morie_tolerance_limits

Distribution-free (Wilks) tolerance limits

Description

Closed-form Wilks (1941) tolerance-interval probability that the sample interval from min(x) to max(x) covers at least `coverage` of the population. Gibbons & Chakraborti Ch 2.11.

Usage

```
morie_tolerance_limits(x, coverage = 0.9, confidence = 0.95)
```

Arguments

`x`	Numeric vector.
`coverage`	Desired population coverage `beta` (default 0.90).
`confidence`	Desired confidence (default 0.95).

Details

P(coverage of (X_(1), X_(n)) >= beta) = 1 - n * beta^(n-1) + (n - 1) * beta^n

Value

Named list: lower, upper, coverage_requested, confidence_achieved, n, method.

References

Wilks (1941); Gibbons & Chakraborti (6e) Ch 2.11.

Examples

```
morie_tolerance_limits(1:100, coverage = 0.90, confidence = 0.95)
```

---

morie_toronto_neighbourhoods

Toronto neighbourhood boundary versions

Description

Wraps loading of the City of Toronto neighbourhood polygon layers for the CURRENT 158-neighbourhood scheme (`HOOD_158`), the HISTORICAL 140-neighbourhood scheme (`HOOD_140`), and the Neighbourhood Improvement Area (NIA) layer. Also provides version-resolution helpers so downstream analyses do not silently mix the two schemes across years.

---

morie_tps_add_hood_140_from_158

Add an equivalent HOOD_140 column to a HOOD_158-keyed data.frame

---

Description

The mirror of `morie_tps_add_hood_158_from_140()`. For 158-hoods that did not exist in the 140-scheme (the newly-split children like Etobicoke City Centre / Islington from 14-Islington-City- Centre-West) the result is the historical parent 140-hood.

Usage

```
morie_tps_add_hood_140_from_158(
  df,
  col_in = NULL,
  col_out = "HOOD_140_equiv",
  crosswalk = NULL
)
```

Arguments

`df`	A TPS crime `data.frame`.
`col_in`	Name of the input HOOD_158 column.
`col_out`	Name of the new column. Default `"HOOD_140_equiv"`.
`crosswalk`	Optional pre-loaded crosswalk; defaults to `morie_to_hood_crosswalk()`.

Value

`df` with the equivalent-code column appended.

`morie_tps_add_hood_158_from_140`

Add an equivalent HOOD_158 column to a HOOD_140-keyed data.frame

---

Description

Looks up each row's `HOOD_140` (or `hood_140` / `NEIGHBOURHOOD_140` / `neighbourhood_140`) in the bundled crosswalk and writes the PRIMARY-overlap 158 hood code into a new column (default name `HOOD_158_equiv`).

Usage

```
morie_tps_add_hood_158_from_140(
  df,
  col_in = NULL,
  col_out = "HOOD_158_equiv",
  crosswalk = NULL
)
```

Arguments

`df`	A TPS crime `data.frame`.
`col_in`	Name of the input HOOD_140 column. By default the first match from `c("HOOD_140", "hood_140", "NEIGHBOURHOOD_140", "neighbourhood_140")` present in `df`.
`col_out`	Name of the new column to add. Default `"HOOD_158_equiv"`.
`crosswalk`	Optional pre-loaded crosswalk; defaults to `morie_to_hood_crosswalk()`.

Details

For 1:1 mappings the result is exact. For splits (1 historical hood -> 2–4 current hoods) the 158 hood with the largest area overlap wins; this is **lossy** – analyses at the 158-level should ideally re-aggregate from the per-incident lat/lon rather than relying on the primary-overlap join.

Value

`df` with the equivalent-code column appended.

Examples

```
df <- data.frame(EVENT_ID = 1:3, HOOD_140 = c("082", "001", "075"))
morie_tps_add_hood_158_from_140(df)
```

---

morie_tps_aggregate_158_to_140

Aggregate per-158 counts to per-140 counts (EXACT for pure cake-cuts)

Description

Cake-cutting in the REVERSE direction. Given a `data.frame` of per- current-hood counts (one row per `hood_158`, one or more numeric count columns), sums across the 158-children of each 140 weighted by `pct_158_in_140 / 100`. For 1:1 hoods the count passes through. For splits the children's counts are summed exactly (each child has `pct_158_in_140 == 100` for clean cake-cuts).

Usage

```
morie_tps_aggregate_158_to_140(
  df,
  hood_158_col = "HOOD_158",
  count_cols = NULL,
  crosswalk = NULL
)
```

Arguments

`df`	A `data.frame` keyed on a 158-hood column.
`hood_158_col`	Name of the 158-hood column. Default `"HOOD_158"`.
`count_cols`	Character vector of numeric count columns. By default every numeric column except `hood_158_col`.
`crosswalk`	Optional pre-loaded crosswalk.

Details

Unlike `morie_tps_disaggregate_140_to_158()`, this requires NO uniform-density assumption when the source is a clean cake-cut – the partition is exhaustive and disjoint by construction. The only lossy case is the `split+merge` edge (one Willowdale East sliver in the bundled OT data); the function handles it via the `pct_158_in_140` weights regardless.

Value

A `data.frame` with one row per 140-hood, columns `hood_140`, `name_140`, and the summed `count_cols`.

Examples

```
df <- data.frame(HOOD_158 = c("167", "168", "001"),
                 incidents = c(40, 60, 42))
morie_tps_aggregate_158_to_140(df)
```

`morie_tps_analyze` *RichResult-emitting analyses for the 13 TPS crime datasets*

Description

R-side port of `morie.tps_all_analyze`. Provides a uniform bundle (temporal + spatial + offence + neighbourhood-concentration) that runs on any of Toronto Police Service's 13 public crime CSVs plus a cross-category comparison driver.

Details

Functions:

- `morie_tps_temporal_summary`: year / month / dow / hour rollups.
- `morie_tps_spatial_summary`: neighbourhood / division / premises / location-type + lat-lon bbox.
- `morie_tps_offence_summary`: OFFENCE / UCR / CSI rollups.
- `morie_tps_neighbourhood_concentration`: Gini + top-10/top-20 share across HOOD_158.
- `morie_tps_crime_compare`: side-by-side counts and YoY across multiple TPS data.frames.
- `morie_tps_analyze_one`: full bundle on one frame.
- `morie_tps_analyze_all`: full bundle across every TPS data.frame supplied in a named list.
- `morie_tps_analyze_assault()`, ..., `morie_tps_analyze_theftover()`: 13 thin convenience aliases.

Each summary callable returns a named `list` carrying summary lines, optional tables, optional warnings, and a plain-language `interpretation`. The aggregate `morie_tps_analyze_one()` nests these sub-results under named keys.

`morie_tps_analyze_all` *Run the full TPS analysis bundle across many TPS data.frames*

Description

Mirrors `morie.tps_all_analyze.analyze_all`. Caller supplies the data.frames; loading from disk is left to the user (R-side loaders live in `R/data_access.R` and `R/dataset_catalog.R`). An optional `out_dir` writes per-dataset `tps_<name>.txt` transcripts.

Usage

```
morie_tps_analyze_all(dfs, out_dir = NULL)
```

Arguments

`dfs`	Named `list` of TPS data.frames keyed by canonical TPS name (e.g. `"Assault"`, `"Homicides"`, ...).
`out_dir`	Optional directory to write per-dataset text dumps. When `NULL`, no files are written.

Value

A named `list` of `morie_tps_result` values, plus a `` `__cross_compare__` `` entry from morie_tps_crime_compare.

`morie_tps_analyze_assault` *Convenience alias: full TPS bundle on the Assault dataset.*

Description

Convenience alias: full TPS bundle on the Assault dataset.

Usage

```
morie_tps_analyze_assault(df)

morie_tps_analyze_autotheft(df)

morie_tps_analyze_bicycletheft(df)

morie_tps_analyze_breakandenter(df)

morie_tps_analyze_communitysafetyindicators(df)
```

```
morie_tps_analyze_hatecrimes(df)

morie_tps_analyze_homicides(df)

morie_tps_analyze_intimatepartnerandfamilyviolence(df)

morie_tps_analyze_neighbourhoodcrimerates(df)

morie_tps_analyze_robbery(df)

morie_tps_analyze_shootingandfirearmdiscarges(df)

morie_tps_analyze_theftfrommovingvehicle(df)

morie_tps_analyze_theftover(df)
```

Arguments

`df`	A TPS Assault data.frame.

Value

A `morie_tps_result`.

`morie_tps_analyze_csi_from_dataframes` *Toronto CSI per-year + per-ward from a named list of TPS data.frames*

Description

High-level orchestration: takes a named list `dfs` of TPS open-data data.frames (one per CSI category) and returns both the per-year and per-ward CSI as a single rich-result object.

Usage

```
morie_tps_analyze_csi_from_dataframes(
  dfs,
  year_col = "OCC_YEAR",
  hood_col = "HOOD_158",
  variant = c("total", "violent")
)
```

Arguments

`dfs`	Named list of TPS data.frames. Keys outside of `MORIE_TPS_CSI_CATEGORIES()` are ignored.
`year_col`	Year column name (default "OCC_YEAR").

`hood_col`	Neighbourhood column name (default "HOOD_158").
`variant`	One of "total" or "violent" (default "total").

Value

A `morie_tps_result` named list carrying `by_year` and `by_hood` data.frames in `payload`.

`morie_tps_analyze_one`

Run the standard TPS analysis bundle on one data.frame

Description

Chains temporal + spatial + offence + concentration into a single nested result. This is the function the 13 convenience aliases (`morie_tps_analyze_assault()`, etc.) wrap.

Usage

```
morie_tps_analyze_one(df, name = "?")
```

Arguments

`df`	A TPS crime data.frame.
`name`	The canonical TPS dataset name (used in titles).

Value

A `morie_tps_result` with named sub-results under `temporal`, `spatial`, `offences`, `concentration`.

`morie_tps_arima_forecast`

ARIMA(1,1,1) monthly-count forecast

Description

Builds a monthly count series via `stats::ts`, fits ARIMA(1,1,1) with `stats::arima`, and forecasts `h` periods ahead with `stats::predict`. AIC is reported from the fit; BIC is computed manually as `AIC + k * (log(n) - 2)`.

Usage

```
morie_tps_arima_forecast(df, h = 12L, ds_name = "?")
```

Arguments

df	A data.frame.
h	Forecast horizon in months.
ds_name	Character label.

Value

A morie_rich_result list with forecast, aic, bic, n_train.

---

morie_tps_assert_hood_version

Assert the HOOD_* schema version of a TPS data.frame

Description

Errors when the expected schema's column is absent. Warns when BOTH schemas are present (downstream code MAY accidentally use the wrong one).

Usage

```
morie_tps_assert_hood_version(df, expected = c("158", "140"))
```

Arguments

df	A TPS crime data.frame.
expected	Either "158" or "140".

Value

Invisibly TRUE on success.

---

morie_tps_available_formats

List of formats this build can actually load.

Description

Always returns csv. excel requires readxl; spatial formats require sf. If a needed namespace isn't installed, that format is omitted from the returned vector.

Usage

```
morie_tps_available_formats()
```

Value

Character vector of available format names, sorted.

`morie_tps_bivariate_moran` *Bivariate Moran's I between two attributes at the same polygons*

Description

Generalises `morie_tps_polygon_morans_i` to two attributes: measures the cross-correlation between attribute X at location i and attribute Y at neighbouring locations j.

Usage

```
morie_tps_bivariate_moran(
  polygons,
  x_col,
  y_col,
  ds_name = "NeighbourhoodCrimeRates",
  k_neighbours = 5L,
  centroid_lat_col = "lat",
  centroid_lon_col = "lon"
)
```

Arguments

`polygons`	An **sf** object, or a data.frame with centroid columns.
`x_col`, `y_col`	The two attributes (column names).
`ds_name`	Tag for the result title.
`k_neighbours`	k for the k-NN weights graph.
`centroid_lat_col`, `centroid_lon_col`	Names of centroid columns when `polygons` is a plain data.frame.

Details

$$I_{xy} = \frac{n}{S_0} \frac{\sum_i \sum_j w_{ij}\, z_i^x\, z_j^y}{\sqrt{\sum_i (z_i^x)^2 \cdot \sum_i (z_i^y)^2}}$$

Polygon centroids and k-NN weights are constructed exactly as in `morie_tps_polygon_morans_i`; distances use the haversine formula for parity with the Python source.

Value

A named list with `I_xy`, `n`, `x_col`, `y_col`.

Examples

```
set.seed(2026)
polys <- data.frame(
  HOOD_ID = letters[1:16],
  lat = rep(43.6 + (0:3) * 0.02, 4),
  lon = rep(-79.4 + (0:3) * 0.02, each = 4),
  ASSAULT_RATE_2024 = rpois(16, 30),
  HOMICIDE_RATE_2024 = rpois(16, 2)
)
morie_tps_bivariate_moran(polys,
  x_col = "ASSAULT_RATE_2024",
  y_col = "HOMICIDE_RATE_2024",
  centroid_lat_col = "lat", centroid_lon_col = "lon")
```

---

morie_tps_bivariate_morans_i

Bivariate Moran's I between two TPS categories

Description

Tests whether category A's per-neighbourhood count co-varies with category B's count in NEIGHBOURING neighbourhoods (spatial spillover). Builds a k-NN row-standardised spatial weights matrix from per-hood centroids derived from category A's WGS84 latitude/longitude. Reports Pearson r alongside as a non-spatial baseline.

Usage

```
morie_tps_bivariate_morans_i(dfs, cat_a, cat_b, k_neighbours = 5L)
```

Arguments

`dfs`	Named list of TPS data.frames keyed by category.
`cat_a`	Name of category A in `dfs`.
`cat_b`	Name of category B in `dfs`.
`k_neighbours`	Number of nearest neighbours per row in W (default `5L`).

Value

A `morie_tps_result` named list.

`morie_tps_category_correlation_matrix`
Pearson correlation across TPS categories' per-hood counts

Description

For every category in `dfs`, computes per-HOOD_158 counts, aligns onto a common (union) hood index, and reports the Pearson correlation matrix.

Usage

```
morie_tps_category_correlation_matrix(dfs)
```

Arguments

`dfs`	Named list of TPS data.frames keyed by category.

Value

A `morie_tps_result` named list with a single correlation table.

`morie_tps_changepoint_detection`
Pettitt-style change-point on yearly incident counts

Description

Implements Pettitt's non-parametric change-point statistic U_t = sum_i sum_j sign(x_i - x_j) for i <= t < j, then reports the year maximising |U_t| and an approximate p-value. No external change-point dependency is required.

Usage

```
morie_tps_changepoint_detection(df, year_col = "OCC_YEAR", ds_name = "?")
```

Arguments

`df`	A `data.frame` with one row per incident.
`year_col`	Year column name.
`ds_name`	Character label.

Value

A `morie_rich_result` list with `changepoint_year`, `K_statistic`, `p_value`, `pre_mean`, `post_mean`.

`morie_tps_compare_hawkes_kernels` *Compare Hawkes models across kernel x baseline combinations*

Description

Fits every supplied (kernel, baseline) combination and ranks by AIC. Mirrors Section 5 of Kwan-Chen-Dunsmuir (2024): the Markovian classical Hawkes is the (exponential, constant) row; the non-Markovian non-stationary models are everything else.

Usage

```
morie_tps_compare_hawkes_kernels(
  df,
  ds_name = "?",
  max_n = 4000L,
  baselines = .TPS_HAWKES_BASELINES,
  kernels = .TPS_HAWKES_KERNELS
)
```

Arguments

`df`	Data frame with `OCC_DATE` or `REPORT_DATE`.
`ds_name`	Dataset name used in titles.
`max_n`	Maximum events to fit.
`baselines`	Baseline kinds to sweep over.
`kernels`	Kernel kinds to sweep over.

Details

Combinations that fail to converge are recorded with an error message rather than aborting the whole comparison.

Value

A `morie_rich_result` with a per-combination summary table, the best (lowest-AIC) combination, and the AIC gap between the classical Markovian model and the winner.

References

Kwan TKJ, Chen F, Dunsmuir WTM (2024). arXiv:2408.09710.

Examples

```
## Not run:
  df <- morie_tps_load_tps_dataset("Assault", nrows = 3000)
  rr <- morie_tps_compare_hawkes_kernels(df, ds_name = "Assault")

## End(Not run)
```

`morie_tps_composite_index`
Composite per-neighbourhood crime-risk index across TPS categories

---

Description

For each TPS category, computes per-HOOD_158 counts, z-standardises across neighbourhoods, and sums (or weight-and-sums) the z-scores to yield a single composite per neighbourhood. Positive composite = neighbourhood with elevated incidence across many crime types; near-zero = average; negative = below-average exposure.

Usage

```
morie_tps_composite_index(dfs, categories = NULL, weights = NULL, top_n = 25L)
```

Arguments

`dfs`	Named list of TPS data.frames keyed by category.
`categories`	Optional character vector restricting categories.
`weights`	Optional named numeric vector of per-category weights; defaults to 1.0 for every loaded category.
`top_n`	How many top/bottom neighbourhoods to surface in the tables (default `25L`).

Value

A `morie_tps_result` named list.

`morie_tps_crime_compare`
Compare counts and trends across multiple TPS categories

Description

Accepts a named list of TPS data.frames (e.g. `list(Assault = df_a, Robbery = df_r, ...)`) and returns a `morie_tps_result` with a total-counts table and (when `OCC_YEAR` is present in every frame) a side-by-side year-by-year matrix.

Usage

```
morie_tps_crime_compare(dfs)
```

Arguments

`dfs`	Named `list` of TPS data.frames.

Value

A `morie_tps_result` list.

`morie_tps_criminal_network_graph`
Premise x neighbourhood co-occurrence network

Description

Builds a co-occurrence network in lieu of the co-offender network from D'Orsogna & Perc (2015) Fig. 9 / Diviak *et al.* (2019). Public TPS data has no co-offender records, so we approximate by projecting (top-N premise types) x (HOOD_158 neighbourhoods) onto a premise-by-premise edge-weighted graph. Edge weight is the count of neighbourhoods in which both premise types appear.

Usage

```
morie_tps_criminal_network_graph(
  category = "Assault",
  sample_rows = 30000L,
  top_n_premises = 20L,
  save_fig = TRUE
)
```

Arguments

`category`	TPS category name.
`sample_rows`	Maximum rows to load.
`top_n_premises`	Number of premise nodes to keep.
`save_fig`	Whether to emit a circular layout PNG.

Value

A `morie_rich_result` with node count, edge count, strongest edge weight, and the adjacency payload.

References

Diviak T, Dijkstra JK, Snijders TAB (2019). Structure, multiplexity, and centrality in a corruption network. *Trends in Organized Crime* 22: 274-297.

Examples

```
## Not run:
  rr <- morie_tps_criminal_network_graph("Assault",
                                         top_n_premises = 10L,
                                         save_fig = FALSE)
  print(rr$summary_lines)

## End(Not run)
```

MORIE_TPS_CSI_CATEGORIES

Canonical CSI category names (the 9 TPS open-data feeds).

Description

Canonical CSI category names (the 9 TPS open-data feeds).

Usage

```
MORIE_TPS_CSI_CATEGORIES()
```

Value

Character vector.

morie_tps_csi_per_neighbourhood

CSI per neighbourhood (HOOD_158)

Description

Mirrors morie_tps_csi_per_year but groups by neighbourhood ID rather than fiscal year. Population is not divided in here because TPS open data does not ship a per-ward population table; callers are expected to merge in the City of Toronto Open Data Neighbourhood-CrimeRates per-ward population for per-capita rates. Returns the un-normalised weighted sum + total count.

Usage

```
morie_tps_csi_per_neighbourhood(
  counts_per_hood,
  variant = c("total", "violent"),
  weights = NULL
)
```

Arguments

counts_per_hood	Long data.frame (columns HOOD_158, category, count) or nested list [[hood]][[category]] = count.
variant	One of "total" or "violent".
weights	Optional override vector of weights.

Value

A data.frame with one row per neighbourhood.

morie_tps_csi_per_year

Toronto CSI per fiscal year from per-category counts

Description

Accepts either a long-format data.frame (columns year, category, count) or a nested list keyed [[year]][[category]] = count.

Usage

```
morie_tps_csi_per_year(
  counts_per_year,
  variant = c("total", "violent"),
  weights = NULL,
  population = NULL,
  per_capita_unit = 100000L,
  rebase_to_year = NULL,
  rebase_to_value = 100
)
```

Arguments

`counts_per_year`	Long data.frame or nested list (see above).
`variant`	One of "total" or "violent".
`weights`	Optional override vector of weights.
`population`	Optional named integer vector (year -> pop); defaults to `MORIE_TPS_TORONTO_POPULATION_BY_Y`
`per_capita_unit`	Rate denominator (default 100000).
`rebase_to_year`	Optional anchor year for the index.
`rebase_to_value`	Index value at the anchor year (default 100).

Details

Returns a data.frame indexed by year with columns:

- `raw_weighted_sum` – sum_c w_c * n_(c,year)
- `total_count` – sum_c n_(c,year)
- `population` – Toronto population that year
- `csi_per_capita` – raw_weighted_sum / population * per_capita_unit
- `simple_count_rate` – total_count / population * per_capita_unit

When `rebase_to_year` is supplied, an additional `csi_index` column is added, anchored so that that year's value equals `rebase_to_value`.

Value

A data.frame with one row per year.

morie_tps_csi_weight *Return the CSI weight for a TPS open-data category.*

Description

Return the CSI weight for a TPS open-data category.

Usage

```
morie_tps_csi_weight(category, variant = c("total", "violent"), weights = NULL)
```

Arguments

`category`	TPS category name (e.g. "Assault", "Homicides").
`variant`	One of "total" or "violent".
`weights`	Optional named numeric vector overriding the built-in tables. When supplied, takes precedence over `variant`.

Value

Numeric scalar (0 if unknown).

morie_tps_data_dir *Default project data directory for TPS open data.*

Description

Resolves to `<repo>/data/datasets/TPS/` when `morie` is loaded out of a source checkout. Users can override per-call via the `path` argument of `morie_tps_load_dataset()`.

Usage

```
morie_tps_data_dir()
```

`morie_tps_dbscan_clusters` *DBSCAN density clusters on lat/long*

Description

Requires the optional **dbscan** package. Coordinates are projected to km via the small-angle latitude factor so `eps_km` is interpretable as a kilometre-scale radius.

Usage

```
morie_tps_dbscan_clusters(
  df,
  ds_name = "?",
  eps_km = 0.25,
  min_samples = 30L,
  max_n = 30000L,
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84"
)
```

Arguments

`df`	Incident-level data.frame.
`ds_name`	Tag for the result title.
`eps_km`	Neighbourhood radius in km.
`min_samples`	DBSCAN `minPts` parameter.
`max_n`	Subsample cap to keep DBSCAN tractable.
`lat_col`, `lon_col`	WGS84 column names.

Value

A named list with the per-cluster table, the count of noise points, and the largest-cluster size.

Examples

```
if (requireNamespace("dbscan", quietly = TRUE)) {
  set.seed(2026)
  df <- data.frame(
    LAT_WGS84 = c(rnorm(60, 43.65, 0.005), rnorm(60, 43.70, 0.005)),
    LONG_WGS84 = c(rnorm(60, -79.40, 0.005), rnorm(60, -79.38, 0.005))
  )
  morie_tps_dbscan_clusters(df, eps_km = 0.5, min_samples = 5L)
}
```

`morie_tps_disaggregate_140_to_158` *Disaggregate per-140 counts to per-158 counts (uniform-density)*

Description

Cake-cutting in the FORWARD direction. Given a `data.frame` of per- historical-hood counts (one row per `hood_140`, one or more numeric count columns), splits each 140's count across its 158-children in proportion to `pct_140_in_158 / 100`. For 1:1 hoods the count is passed through unchanged. For splits the count is partitioned (e.g. 140-75 Church-Yonge Corridor's 100 incidents become 59.24 in 158-168 Downtown Yonge East and 40.76 in 158-167 Church-Wellesley).

Usage

```
morie_tps_disaggregate_140_to_158(
  df,
  hood_140_col = "HOOD_140",
  count_cols = NULL,
  crosswalk = NULL
)
```

Arguments

`df`	A `data.frame` keyed on a 140-hood column.
`hood_140_col`	Name of the 140-hood column. Default `"HOOD_140"`.
`count_cols`	Character vector of numeric count columns to disaggregate. Default: every numeric column in `df` except `hood_140_col`.
`crosswalk`	Optional pre-loaded crosswalk; defaults to `morie_to_hood_crosswalk()`.

Details

This assumes UNIFORM SPATIAL DENSITY of the underlying events inside each 140-hood – which is the best you can do without per- incident lat/lon. If you have lat/lon, prefer re-binning from points (via `sf::st_join` against `morie_to_neighbourhoods("158", offline = FALSE)`) over this uniform-density approximation.

Value

A `data.frame` with columns `hood_158`, `name_158`, `hood_140`, all chosen `count_cols` (now per-158 fractional counts), and `pct_140_in_158` (the cake-cut weight applied).

Examples

```
df <- data.frame(HOOD_140 = c("075", "001"),
                 incidents = c(100, 42))
morie_tps_disaggregate_140_to_158(df)
```

`morie_tps_district_for_centroid`
Return the pre-1998 borough name for a (lat, lon) centroid

Description

Toronto amalgamated in 1998; the six former municipalities (Etobicoke, North York, Scarborough, York, Old Toronto, East York) are still in common use for district-level reporting and are bbox-defined here.

Usage

```
morie_tps_district_for_centroid(lat, lon)
```

Arguments

`lat`	Numeric latitude (WGS84).
`lon`	Numeric longitude (WGS84).

Value

A character scalar. Defaults to `"Old Toronto"` when no bbox matches.

`morie_tps_fetch_category`
Fetch one TPS category as a CSV, paging until exhausted.

Description

Walks the ArcGIS REST `/query` endpoint for the category's FeatureServer layer, accumulates all features in memory, and writes a single CSV file to `cache_dir`. Returns the CSV path.

Usage

```
morie_tps_fetch_category(
  category,
  cache_dir = NULL,
  where = "1=1",
  overwrite = FALSE,
  max_records_per_page = 2000L
)
```

Arguments

`category`	One of `morie_tps_list_categories()`.
`cache_dir`	Directory to write the CSV into. Defaults to `tempdir()` (CRAN-safe; the Python default of `~/.cache/morie/tps` requires `R_user_dir` opt-in in R).
`where`	ArcGIS SQL `where` clause (default `"1=1"`).
`overwrite`	If `FALSE` and the output exists, return it without re-downloading.
`max_records_per_page`	Pagination size (server caps at 2000).

Value

Path to the written CSV file.

`morie_tps_fetch_dataframe` *Fetch a TPS category and return it as a* `data.frame`.

Description

Thin wrapper over `morie_tps_fetch_category()`: writes the CSV then reads it back. Mirrors the Python `fetch_tps_dataframe` convenience used as a `DATASET_CATALOG` fetcher.

Usage

```
morie_tps_fetch_dataframe(category, ...)
```

Arguments

`category`	One of `morie_tps_list_categories()`.
`...`	Passed through to `morie_tps_fetch_category()`.

Value

A `data.frame`.

`morie_tps_fokker_planck_grid`
1-D Fokker-Planck density evolution under OU drift+diffusion

Description

Fits the OU parameters (theta, mu, sigma) on daily counts (same OLS-on-first-differences as `morie_tps_langevin_simulate`), then evolves an initial gaussian density centred on the last observation by an explicit advection-diffusion finite-difference scheme with reflective boundaries on a grid spanning from 0 to 1.5 * max(counts) + 1.

Usage

```
morie_tps_fokker_planck_grid(df, ds_name = "?", n_grid = 64L, n_steps = 200L)
```

Arguments

`df`	A `data.frame`.
`ds_name`	Character label.
`n_grid`	Grid points (default 64).
`n_steps`	Time steps (default 200, each of length 0.05 days).

Value

A `morie_rich_result` list with `theta`, `mu`, `sigma`, `grid`, `density`, `stationary_peak`.

`morie_tps_getis_ord_g_star`
Local Getis-Ord Gi statistic per neighbourhood*

Description

Returns Gi* per neighbourhood (count vector aggregated from the incident data.frame), using a binary k-NN spatial weights matrix with self-inclusion (Gi* convention). z-score interpretation: Gi*1.96 = significant hot spot at alpha=0.05; Gi* < -1.96 =significant cold spot.

Usage

```
morie_tps_getis_ord_g_star(
  df,
  ds_name = "?",
  hood_col = "HOOD_158",
  k_neighbours = 5L,
  top_n = 20L,
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84"
)
```

Arguments

`df`	Incident-level data.frame.
`ds_name`	Tag for the result title.
`hood_col`	Neighbourhood id column.
`k_neighbours`	k for the (binary) k-NN weights graph.
`top_n`	Number of top hot/cold spots to surface.
`lat_col`, `lon_col`	WGS84 column names.

Value

A named list with Gi* per hood, the top hot/cold spot tables, and tallies of hot/cold spots at alpha=0.05.

Examples

```
set.seed(2026)
df <- data.frame(
  HOOD_158 = sample(letters[1:20], 400, replace = TRUE),
  LAT_WGS84 = 43.6 + runif(400, 0, 0.2),
  LONG_WGS84 = -79.4 + runif(400, 0, 0.2)
)
morie_tps_getis_ord_g_star(df)
```

`morie_tps_gini_concentration` *Gini coefficient of a numeric vector*

Description

G=0 perfectly even, G=1 perfectly concentrated. Used for the neighbourhood-concentration callable below; exposed because tests and downstream code may want it directly.

Usage

```
morie_tps_gini_concentration(x)
```

Arguments

`x`	Numeric vector (e.g. per-spatial-unit incident counts).

Value

A scalar Gini coefficient in `[0, 1]` (or NA when input is empty).

`morie_tps_hawkes_advanced`
Non-stationary Hawkes with non-exponential kernels (R port)

Description

R port of `morie.tps_hawkes_advanced`. Implements the Kwan, Chen and Dunsmuir (2024, arXiv:2408.09710v1) methodology for Hawkes process likelihood inference when the baseline intensity is time-varying *and* the excitation kernel is non-exponential (so the intensity process is non-Markovian).

Details

The complete intensity is

$$\lambda(t) = (t) + \int_0^{t-} g(t-s)\, dN_s,$$

with kernel decomposition $g(u) = \eta \cdot \tilde{g}(u)$ where $\eta \in (0,1)$ is the branching ratio (mean offspring per event) and $\tilde{g}$ is a probability density on $[0,\infty)$. Stationarity requires $\eta < 1$.

Supported kernels: exponential, gamma, Weibull, Lomax (Pareto-II). Supported baselines: constant and sinusoidal-with-trend

$$(t) = \exp\big(a_0 + a_1(t/T) + a_2 \sin(2\pi t/365.25) + a_3 \cos(2\pi t/365.25)\big).$$

Companion to `morie_tps_hawkes_temporal_fit` (exponential / constant Markovian special case) in `morie.tps_stochastic`.

Goodness-of-fit uses time-rescaling residuals (Brown *et al.* 2002 *Neural Comput.* 14: 325-346) and a Kolmogorov-Smirnov test against Uniform(0,1).

If the optional R package **hawkes** or **emhawkes** is installed it is consulted for the exponential-kernel constant- baseline fast path; otherwise the negative log-likelihood is computed in base R via direct O(n^2) summation. The non-Markovian kernels (gamma, Weibull, Lomax) always use the base-R path – those kernels lack the memorylessness required for O(n) recursion.

Functions:

- `morie_tps_hawkes_advanced_fit` – fit one (kernel, baseline) combination and produce a rich result with time-rescaling KS diagnostics.
- `morie_tps_compare_hawkes_kernels` – 8-way AIC comparison across (kernel, baseline) combinations.
- `morie_tps_hawkes_markovian_vs_nonmarkovian` – focused 2x2 comparison: classical exp/const vs gamma/ sinusoidal.

References

Kwan TKJ, Chen F, Dunsmuir WTM (2024). Likelihood inference for non-stationary Hawkes processes. arXiv:2408.09710v1.

Brown EN, Barbieri R, Ventura V, Kass RE, Frank LM (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation* 14: 325-346.

Mohler GO, Short MB, Brantingham PJ, Schoenberg FP, Tita GE (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association* 106: 100-108.

`morie_tps_hawkes_advanced_fit` *Fit a single (kernel, baseline) Hawkes specification*

Description

Companion to the Markovian exponential / constant fit in `morie_tps_hawkes_temporal_fit`. Supports the four kernels (exponential, gamma, Weibull, Lomax) and two baselines (constant, sinusoidal) of Kwan-Chen-Dunsmuir (2024).

Usage

```
morie_tps_hawkes_advanced_fit(
  df,
  kernel = "gamma",
  baseline = "sinusoidal",
  ds_name = "?",
  max_n = 5000L
)
```

Arguments

`df`	A data frame with an `OCC_DATE` or `REPORT_DATE` column.
`kernel`	Excitation kernel: one of `"exponential"`, `"gamma"`, `"weibull"`, `"lomax"`.
`baseline`	Baseline kind: `"constant"` or `"sinusoidal"`.
`ds_name`	Dataset name used in titles and warnings.
`max_n`	Maximum number of events to retain (for tractable O(n^2) MLE on the non-Markovian path).

Details

If the optional packages **hawkes** or **emhawkes** are available the (exponential, constant) special case can delegate to their compiled likelihood routines; the non-Markovian kernels always use the base-R O(n^2) negative log-likelihood with L-BFGS-B optimisation under explicit box constraints.

Goodness-of-fit is reported via time-rescaling residuals (Brown *et al.* 2002) and a Kolmogorov-Smirnov test against Uniform(0, 1).

Value

A `morie_rich_result` with branching ratio, stationarity verdict, kernel and baseline parameters, negative log-likelihood, AIC, BIC, and time-rescaling KS statistic.

References

Kwan TKJ, Chen F, Dunsmuir WTM (2024). Likelihood inference for non-stationary Hawkes processes. arXiv:2408.09710v1.

Examples

```
## Not run:
  df <- morie_tps_load_tps_dataset("Assault", nrows = 4000)
  rr <- morie_tps_hawkes_advanced_fit(df, kernel = "gamma",
                                      baseline = "sinusoidal",
                                      ds_name = "Assault")
  print(rr$summary_lines)

## End(Not run)
```

`morie_tps_hawkes_markovian_vs_nonmarkovian`

Focused 2x2 Markovian vs non-Markovian Hawkes comparison

Description

Fits the four (kernel, baseline) combinations corresponding to the two endpoints of the Kwan-Chen-Dunsmuir framework: classical exponential / constant Markovian, against gamma / sinusoidal non-Markovian. Faster to run than the full 8-way comparison and suitable for dashboard surfaces.

Usage

```
morie_tps_hawkes_markovian_vs_nonmarkovian(df, ds_name = "?", max_n = 4000L)
```

Arguments

`df`	Data frame with `OCC_DATE` or `REPORT_DATE`.
`ds_name`	Dataset name used in titles.
`max_n`	Maximum events to fit.

Value

A `morie_rich_result` from `morie_tps_compare_hawkes_kernels` restricted to the 2x2 sub-grid.

References

Kwan TKJ, Chen F, Dunsmuir WTM (2024). arXiv:2408.09710.

Examples

```
## Not run:
  df <- morie_tps_load_tps_dataset("Assault", nrows = 2000)
  rr <- morie_tps_hawkes_markovian_vs_nonmarkovian(df,
                                                   ds_name = "Assault")

## End(Not run)
```

---

morie_tps_hawkes_temporal_fit

Temporal exponential-kernel Hawkes fit

Description

Maximum-likelihood fit of a temporal-only exponential Hawkes process to incident times. Optimisation runs in base R (`stats::optim`, Nelder-Mead). Reports background rate mu, branching ratio kappa, decay omega, and the AIC / BIC of the fit.

Usage

```
morie_tps_hawkes_temporal_fit(df, ds_name = "?", max_n = 5000L)
```

Arguments

`df`	A `data.frame` with TPS-shaped date columns.
`ds_name`	Character label for the dataset.
`max_n`	Maximum number of incident times to fit (random subsample seeded with 42 if exceeded).

Value

A `morie_rich_result` list with `mu`, `kappa`, `omega`, `branching`, `nll`, `aic`, `bic`.

`morie_tps_inspection_game_phase` *Helbing-Szolnoki inspection-game phase diagram*

Description

Three-strategy replicator dynamics (cooperator C, defector / predator P, punisher / inspector O) swept across a grid in the (temptation T, inspection cost gamma) plane. Each grid point runs the replicator update to steady state and records the defector share as a "crime rate" proxy. Reproduces the qualitative phase diagram from D'Orsogna & Perc (2015) sec. 5 / Fig. 8.

Usage

```
morie_tps_inspection_game_phase(
  n_temptations = 20L,
  n_costs = 20L,
  n_steps = 600L,
  save_fig = TRUE
)
```

Arguments

`n_temptations`, `n_costs`	Grid resolution.
`n_steps`	Replicator iterations per grid point.
`save_fig`	Whether to write the phase-diagram PNG.

Value

A `morie_rich_result` containing the mean, min, max steady-state defector frequency across the grid, plus the resolution and step count used.

References

Helbing D, Szolnoki A, Perc M, Szabo G (2010). Punish, but not too hard. *New Journal of Physics* 12: 083005.

Examples

```
## Not run:
  rr <- morie_tps_inspection_game_phase(
    n_temptations = 8L, n_costs = 8L, n_steps = 120L,
    save_fig = FALSE)
  print(rr$summary_lines)

## End(Not run)
```

`morie_tps_kde_density` *2-D kernel density estimate of geocoded incidents*

Description

Evaluates a Gaussian KDE on incident lat/long and returns summary statistics plus the (lat, lon) of the densest observation. Prefers **MASS**`::kde2d` when available; otherwise uses a pure base-R Gaussian kernel evaluated at the observation points (i.e. kernel density at each datum).

Usage

```
morie_tps_kde_density(
  df,
  bandwidth = 0.005,
  ds_name = "?",
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84"
)
```

Arguments

`df`	Incident-level data.frame.
`bandwidth`	Bandwidth multiplier passed to the 2-D KDE; see `MASS::kde2d`'s `h` argument when MASS is available. In the base-R fallback this is the Gaussian sigma (in degrees).
`ds_name`	Tag for the result title.
`lat_col`, `lon_col`	WGS84 column names.

Value

A named list with summary stats including max/mean/median density and the (lat, lon) of the densest observation.

Examples

```
set.seed(2026)
df <- data.frame(
  LAT_WGS84 = 43.6 + rnorm(120, 0, 0.05),
  LONG_WGS84 = -79.4 + rnorm(120, 0, 0.05)
)
morie_tps_kde_density(df, bandwidth = 0.01)
```

morie_tps_langevin_simulate

Euler-Maruyama Ornstein-Uhlenbeck simulation

Description

Fits an OU process $dX_t = theta(mu - X_t)dt + sigmadW_t$ to daily incident counts via OLS on first-differences, then runs `n_paths` forward simulations of length `T_days`.

Usage

```
morie_tps_langevin_simulate(
  df,
  ds_name = "?",
  n_paths = 100L,
  T_days = 365L,
  dt = 1,
  seed = 42L
)
```

Arguments

`df`	A `data.frame`.
`ds_name`	Character label.
`n_paths`	Number of forward paths to simulate.
`T_days`	Forward horizon in days.
`dt`	Time step (days).
`seed`	RNG seed.

Value

A `morie_rich_result` list with `theta`, `mu`, `sigma`, `paths` (matrix of n_paths x n_steps), and final-day quantiles.

morie_tps_layer_urls *TPS ArcGIS layer URLs known to MORIE*

Description

Each layer's `/query` endpoint is appended at request time. Folded into the `morie_tps_layer_urls()` Rd via `@rdname` to avoid the case-insensitive filesystem collision between `MORIE_TPS_LAYER_URLS.Rd` and `morie_tps_layer_urls.Rd`.

Usage

```
morie_tps_layer_urls()

MORIE_TPS_LAYER_URLS
```

Format

An object of class `character` of length 9.

Value

Named character vector mapping TPS category names to ArcGIS FeatureServer layer roots.

Examples

```
urls <- morie_tps_layer_urls()
names(urls) # categories: Assault, AutoTheft, Homicide, ...
length(urls) # number of layers
```

---

morie_tps_levy_flight_alpha

Levy-flight tail exponent on consecutive-incident steps

Description

Computes the Hill maximum-likelihood estimator of the upper-tail Pareto exponent α of the step-length distribution between chronologically consecutive incidents, following Brockmann, Hufnagel & Geisel (2006). For a power-law tail $p(\ell) \propto \ell^{-\alpha}$ on $\ell \geq \ell_{\min}$ the Hill MLE is

$$\hat{\alpha} = 1 + n / \sum_i \log(\ell_i / \ell_{\min}).$$

Standard error is obtained by 200 nonparametric bootstrap resamples.

Usage

```
morie_tps_levy_flight_alpha(
  category = "Assault",
  sample_rows = 30000L,
  lmin_km = 0.5,
  save_fig = TRUE
)
```

Arguments

`category`	TPS category name.
`sample_rows`	Maximum rows to load.
`lmin_km`	Lower tail cutoff in km.
`save_fig`	Whether to emit a log-log empirical-vs-fit PNG.

Value

A `morie_rich_result` with $\hat{\alpha}$, bootstrap SE, sample-size diagnostics, and a Lévy-regime interpretation.

References

Brockmann D, Hufnagel L, Geisel T (2006). The scaling laws of human travel. *Nature* 439: 462-465.

Examples

```
## Not run:
  rr <- morie_tps_levy_flight_alpha("Assault", save_fig = FALSE)
  print(rr$summary_lines$alpha)

## End(Not run)
```

`morie_tps_list_categories` *List TPS categories known to the fetcher.*

Description

List TPS categories known to the fetcher.

Usage

```
morie_tps_list_categories()
```

Value

Character vector of category names, sorted.

`morie_tps_list_datasets` *List all TPS datasets as a* `data.frame`.

Description

Returns one row per registered category with columns `name`, `description`, and `primary_date`.

Usage

```
morie_tps_list_datasets()
```

Value

A `data.frame` sorted by `name`.

Examples

```
morie_tps_list_datasets()
```

`morie_tps_list_formats` *Map TPS format name -> path of the file that would be loaded.*

Description

Formats whose sibling directory or file is not present on disk are omitted from the returned named character vector. Use this to discover which formats a given category actually exports.

Usage

```
morie_tps_list_formats(name)
```

Arguments

`name`	TPS category. Case-insensitive.

Value

Named character vector (`format` -> file path).

`morie_tps_load` *Load TPS dataset* `name` *in the given* `format`*.*

Description

Mirror of Python's `morie.tps_io.load_tps`. `csv` and `excel` work with base R / `readxl`; all spatial formats (`geojson`, `featurecollection`, `kml`, `geopackage`, `sqlitegeodatabase`, `shapefile`, `filegeodatabase`) are gated behind `requireNamespace("sf")` and surface a clean install message if the dependency is missing.

Usage

```
morie_tps_load(name, format = "csv", nrows = NULL)
```

Arguments

name	TPS category. Case-insensitive.
format	One of MORIE_TPS_SUPPORTED_FORMATS.
nrows	Optional integer cap on rows.

Value

A data.frame (spatial readers return the dropped-sf data frame; geometry column is preserved as an sfc).

---

morie_tps_load_dataset

Load one TPS dataset by category name (CSV thin path).

Description

name is case-insensitive. Pass nrows = N for a quick sample while developing against the largest tables.

Usage

```
morie_tps_load_dataset(name, path = NULL, csv_filename = NULL, nrows = NULL)
```

Arguments

name	Character scalar. One of names(MORIE_TPS_REGISTRY), case-insensitive.
path	Optional character scalar. Override the CSV file or directory to load from. If a directory, the first *.csv inside is picked. If NULL, the loader walks morie_tps_data_dir().
csv_filename	Optional filename inside the category's CSV/ directory.
nrows	Optional integer. Cap on rows to load.

Details

For non-CSV sibling formats (Excel, GeoJSON, KML, GeoPackage, Shapefile, etc.), use morie_tps_load() from tps_io.R instead.

Value

A data.frame (the CSV contents) with tolerant OCCURRENCE_* / REPORTED_* column renaming applied.

Examples

```
## Not run:
df <- morie_tps_load_dataset("Assault", nrows = 1000L)

## End(Not run)
```

`morie_tps_local_morans_i` *LISA – local Moran's Ii per neighbourhood*

Description

Computes local Moran's Ii for each neighbourhood given a k-NN spatial weights graph on centroid lat/long, with HH / LL / HL / LH quadrant classification.

Usage

```
morie_tps_local_morans_i(
  df,
  hood_col = "HOOD_158",
  ds_name = "?",
  k_neighbours = 5L,
  top_n = 20L,
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84"
)
```

Arguments

`df`	Incident-level data.frame.
`hood_col`	Neighbourhood id column.
`ds_name`	Tag for the result title.
`k_neighbours`	k for the spatial weights graph.
`top_n`	Number of top-Ii rows to surface in the result table.
`lat_col`, `lon_col`	WGS84 column names.

Value

A named list with `table` (data.frame of per-hood I_i, z, Wz, quadrant) and quadrant tallies.

Examples

```
set.seed(2026)
df <- data.frame(
  HOOD_158 = sample(letters[1:15], 300, replace = TRUE),
  LAT_WGS84 = 43.6 + runif(300, 0, 0.2),
  LONG_WGS84 = -79.4 + runif(300, 0, 0.2)
)
morie_tps_local_morans_i(df, top_n = 5L)
```

`morie_tps_lotka_volterra_police_crime` *Lotka-Volterra predator-prey on yearly crime counts*

Description

Treats yearly category counts as the prey $x(t)$ and a 3-year rolling mean as a placeholder predator $y(t)$ (TPS does not yet expose a public mass-stop / use-of-force time series). Under the classical Lotka-Volterra system,

$$\dot{x} = \alpha x - \beta xy, \quad \dot{y} = \delta xy - \gamma y,$$

the small-amplitude oscillation around the equilibrium has period $T = 2\pi/\sqrt{\alpha\gamma}$. Growth rate α is estimated from log-differences of `x`; γ symmetrically from `y`; the interaction rates β, δ follow by the equilibrium relations.

Usage

```
morie_tps_lotka_volterra_police_crime(category = "Assault", save_fig = TRUE)
```

Arguments

`category`	TPS category name.
`save_fig`	Whether to write a yearly time-series PNG.

Value

A `morie_rich_result` with the four LV parameters, the linearised cycle period, the year range, and a qualitative interpretation.

References

D'Orsogna MR, Perc M (2015). Statistical physics of crime: A review. *Physics of Life Reviews* 12: sec. 3.4.

Examples

```
## Not run:
  rr <- morie_tps_lotka_volterra_police_crime("Assault",
                                              save_fig = FALSE)
  print(rr$summary_lines)

## End(Not run)
```

`morie_tps_moran_sweep_heatmap` *Sweep polygon Moran's I across (category x year)*

Description

Loops `morie_tps_polygon_morans_i` over a grid of value-column prefixes and years, returning the resulting matrix of Moran's I values for downstream visualisation as a heatmap.

Usage

```
morie_tps_moran_sweep_heatmap(
  polygons,
  category_prefixes = NULL,
  years = NULL,
  k_neighbours = 5L,
  ds_name = "NeighbourhoodCrimeRates",
  centroid_lat_col = "lat",
  centroid_lon_col = "lon"
)
```

Arguments

`polygons`	An **sf** object or data.frame with centroid columns and per-year value columns.
`category_prefixes`	Character vector of column prefixes. Defaults to the 9 published TPS rate categories.
`years`	Integer vector of years. Defaults to 2014:2024.
`k_neighbours`	k for the k-NN weights graph passed down.
`ds_name`	Tag for the result title.
`centroid_lat_col`, `centroid_lon_col`	Centroid column names forwarded to `morie_tps_polygon_morans_i`.

Details

Column names are constructed as `paste0(prefix, "_", year)`.

Value

A named list with the (category x year) Moran's I matrix.

Examples

```
set.seed(2026)
polys <- data.frame(
  HOOD_ID = letters[1:16],
  lat = rep(43.6 + (0:3) * 0.02, 4),
  lon = rep(-79.4 + (0:3) * 0.02, each = 4),
  ASSAULT_RATE_2023 = rpois(16, 30),
  ASSAULT_RATE_2024 = rpois(16, 32),
  HOMICIDE_RATE_2023 = rpois(16, 2),
  HOMICIDE_RATE_2024 = rpois(16, 2)
)
morie_tps_moran_sweep_heatmap(polys,
  category_prefixes = c("ASSAULT_RATE", "HOMICIDE_RATE"),
  years = c(2023L, 2024L),
  centroid_lat_col = "lat", centroid_lon_col = "lon")
```

---

morie_tps_morans_i_neighbourhood

Global Moran's I on neighbourhood-level incident counts

Description

Builds a k-NN spatial weights matrix from neighbourhood centroids (mean LAT/LONG of incidents in each hood) and computes the global Moran's I on the count vector. The Cliff-Ord normal-assumption variance is used for the z-score and two-sided p-value.

Usage

```
morie_tps_morans_i_neighbourhood(
  df,
  hood_col = "HOOD_158",
  ds_name = "?",
  k_neighbours = 5L,
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84",
  use_spdep = FALSE
)
```

Arguments

`df`	Incident-level data.frame.
`hood_col`	Character. Neighbourhood id column (default `"HOOD_158"`).
`ds_name`	Character. Tag for the result title.
`k_neighbours`	k for the k-NN spatial weights graph (default 5).
`lat_col`, `lon_col`	WGS84 column names (default `"LAT_WGS84"` / `"LONG_WGS84"`).
`use_spdep`	If `TRUE` and **spdep** is installed, delegate the test to `spdep::moran.test` (with a row- standardised listw). Default `FALSE`.

Value

A named list with classes `morie_tps_spatial_result`, `morie_rich_result`, `list`. Numeric outputs include `moran_I`, `expected_I`, `var_I`, `z_score`, `p_value`, `n`.

Examples

```
set.seed(2026)
n_inc <- 400
df <- data.frame(
  HOOD_158 = sample(letters[1:20], n_inc, replace = TRUE),
  LAT_WGS84 = 43.6 + runif(n_inc, 0, 0.2),
  LONG_WGS84 = -79.4 + runif(n_inc, 0, 0.2)
)
morie_tps_morans_i_neighbourhood(df)
```

---

morie_tps_neighbourhood_concentration

How concentrated is crime across Toronto's 158 neighbourhoods?

Description

Uses HOOD_158 (the 158-neighbourhood scheme) and reports a Gini coefficient plus the cumulative share of incidents in the top-10 and top-20 neighbourhoods.

Usage

```
morie_tps_neighbourhood_concentration(df, ds_name = "?")
```

Arguments

`df`	A TPS crime data.frame.
`ds_name`	Optional dataset label used in the result title.

Value

A `morie_tps_result` list with `payload$gini`, `payload$n_hoods`, `payload$p_top10`, `payload$p_top20`.

`morie_tps_offence_summary`
Offence-distribution rollup for a TPS crime data.frame

Description

Top-20 OFFENCE / UCR_CODE / CSI_CATEGORY tables (whichever are present).

Usage

```
morie_tps_offence_summary(df, ds_name = "?")
```

Arguments

`df`	A TPS crime data.frame.
`ds_name`	Optional dataset label used in the result title.

Value

A `morie_tps_result` named list.

`morie_tps_polygon_morans_i`
Polygon-aware Moran's I on a value column

Description

Accepts an **sf** object (recommended) carrying neighbourhood polygons and a numeric value column, computes polygon centroids via `sf::st_centroid`, then runs Moran's I with a k-NN row- standardised weights matrix on those centroids. Falls back to a `data.frame` carrying precomputed centroid columns when **sf** is unavailable.

Usage

```
morie_tps_polygon_morans_i(
  polygons,
  value_col,
  ds_name = "NeighbourhoodCrimeRates",
  k_neighbours = 5L,
  centroid_lat_col = "lat",
  centroid_lon_col = "lon"
)
```

Arguments

polygons	An **sf** object, or a data.frame with centroid columns.
value_col	Column to test for spatial autocorrelation.
ds_name	Tag for the result title.
k_neighbours	k for the k-NN weights graph.
centroid_lat_col, centroid_lon_col	Names of the centroid columns when polygons is a plain data.frame.

Value

A named list with moran_I, z_score, p_value, n.

Examples

```
set.seed(2026)
polys <- data.frame(
  HOOD_ID = letters[1:16],
  lat = rep(43.6 + (0:3) * 0.02, 4),
  lon = rep(-79.4 + (0:3) * 0.02, each = 4),
  ASSAULT_RATE_2024 = rpois(16, 30)
)
morie_tps_polygon_morans_i(polys, value_col = "ASSAULT_RATE_2024",
  centroid_lat_col = "lat", centroid_lon_col = "lon")
```

---

morie_tps_pretty_label

Convert a SQL-style column name to a prose label

Description

ASSAULT_RATE_2024 -> "Assault rate * 2024". Strips underscores and casing so plot titles look like prose.

Usage

```
morie_tps_pretty_label(s)
```

Arguments

s	Character scalar (column name).

Value

Character scalar (display label).

morie_tps_project_xy *Project (lat, lon) to rotated planar kilometres*

Description

Equirectangular projection centred at (lat_c, lon_c), then rotated `rot_deg_cw` degrees clockwise (default 17.5). Returns kilometres east-of-centre (after rotation) and kilometres north-of-centre (after rotation).

Usage

```
morie_tps_project_xy(
  lat,
  lon,
  rot_deg_cw = .MORIE_TPS_ROT_DEG_CW,
  lat_c = .MORIE_TPS_LAT_C,
  lon_c = .MORIE_TPS_LON_C
)
```

Arguments

`lat`	Numeric vector of latitudes (WGS84 degrees).
`lon`	Numeric vector of longitudes (WGS84 degrees).
`rot_deg_cw`	Clockwise rotation in degrees (default 17.5).
`lat_c`, `lon_c`	Centre-point of the projection (default downtown Toronto: 43.70 N, 79.40 W).

Details

Clockwise convention: positive `rot_deg_cw` rotates the map so a line that previously sloped up-right slopes less (or down-right).

Value

A named list with numeric vectors `x` (km east of centre) and `y` (km north of centre).

morie_tps_psdp_layers *List the TPS PSDP layers wrapped by morie*

Description

List the TPS PSDP layers wrapped by morie

Usage

```
morie_tps_psdp_layers()
```

Value

A `data.frame` with columns `layer_key`, `label`, `arcgis_url`, `fixture`, `hub_id` (3TT+ canonical id matching the TPS Hub catalog).

MORIE_TPS_REGISTRY *Registry of TPS open-data categories.*

Description

A named list of one-row metadata records keyed by canonical category name. Each entry holds `description`, `primary_date` (canonical date column name), and `has_geometry` (whether LAT/LONG WGS84 columns are expected).

Usage

```
MORIE_TPS_REGISTRY
```

Format

An object of class `list` of length 13.

---

morie_tps_render *Toronto Police Service map rendering (R-side)*

Description

R-side port of `morie.tps_render`. Carries the two design rules from the Python module (per the author, 2026-05-07):

Details

1. No floating neighbourhood text labels on the map – hot-spot identification is delivered via the `morie_tps_*` result tables, not via on-canvas text.
2. Map is rotated approximately 17.5 degrees clockwise in projected space so Lake Ontario's shoreline sits level horizontally – matching the Sigar Li 2022 "Hotspot Policing for the City of Toronto" poster aesthetic and the Hohl 2024 ALMI homicide-cluster map.

Plotting back-ends are gated behind **ggplot2**; without it, the callables fall back to base `plot()`. Heavy panels (kernel-density, LISA, Getis-Ord, Kulldorff scan) that depend on the Python TPS spatial modules are intentionally not ported here: the projection + base choropleth / point-pattern primitives below are enough for the empirical paper's figures.

Functions:

- morie_tps_project_xy: degrees -> rotated planar km.
- morie_tps_pretty_label: ASSAULT_RATE_2024 -> "Assault rate * 2024".
- morie_tps_district_for_centroid: lat/lon -> pre-1998 borough name.
- morie_tps_render_choropleth: polygon choropleth (ggplot2 if available, else base R).
- morie_tps_render_points: point-pattern map (incident dots + optional DBSCAN colouring).
- morie_tps_render_yearly_grid: small-multiples by year.

`morie_tps_render_choropleth` *Polygon choropleth map for Toronto (158 wards)*

Description

Renders a sequential-colour choropleth from a polygon data.frame carrying one row per neighbourhood with a list-column of WGS84 (lon, lat) rings (`geometry`) and a numeric `rate_col`. This signature deliberately matches what `rmorie::morie_fetch("https://.../NeighbourhoodCrime = "geojson")` returns once unrolled.

Usage

```
morie_tps_render_choropleth(
  polys,
  rate_col = "ASSAULT_RATE_2024",
  title = NULL,
  cmap = "YlOrRd",
  outfile = NULL,
  fig_w = 12,
  fig_h = 7,
  show_ids = TRUE,
  border_color = "#1a1a1a",
  border_lw = 0.7
)
```

Arguments

polys	A `data.frame` or `tibble` with columns `geometry` (list-column of N x 2 lon/lat matrices) and the metric column named in `rate_col`. An optional `HOOD_ID` or `AREA_ID` column drives the show-IDs labels.
rate_col	Name of the metric column. Default `"ASSAULT_RATE_2024"`.
title	Plot title; defaults to a Hohl-2024-style auto-label.
cmap	Sequential colour palette name (default `"YlOrRd"`).

`outfile`	Path to write the image (`.png`, `.pdf`, `.svg`). When `NULL`, the function returns the plot object without writing.
`fig_w`, `fig_h`	Figure size in inches.
`show_ids`	When `TRUE` (default), draw small numeric polygon-ID labels at each ward centroid.
`border_color`	Polygon edge colour.
`border_lw`	Polygon edge linewidth.

Details

When **ggplot2** is available the render uses `geom_polygon` + `scale_fill_distiller`; otherwise the base R `polygon()` primitive is used.

Value

A `ggplot` object (when ggplot2 is loaded) or `invisible(NULL)` for the base-R fallback; the file path is returned invisibly when `outfile` is supplied.

`morie_tps_render_dbscan` *DBSCAN cluster figure on TPS-projected points*

Description

Runs DBSCAN on rotated-km coordinates and colours points by cluster label, with noise rendered grey. Requires the suggested **dbscan** package; without it a base-graphics single-colour fallback is drawn.

Usage

```
morie_tps_render_dbscan(
  points_df,
  eps_km = 0.5,
  min_samples = 8L,
  outfile = NULL,
  ...
)
```

Arguments

`points_df`	data.frame with columns `lat` / `lon` (or `LAT_WGS84` / `LONG_WGS84`).
`eps_km`	DBSCAN epsilon in kilometres.
`min_samples`	Minimum samples per cluster.
`outfile`	Optional output path.
`...`	Extra plotting args (size, alpha, palette).

Value

ggplot object or invisible NULL.

`morie_tps_render_district_proportional`
District-level proportional-symbol map

Description

Renders one centroid-anchored circle per polygon row, sized proportionally to `count_col`. Useful for showing per-district incident counts without colour-coding the polygons themselves.

Usage

```
morie_tps_render_district_proportional(
  polys,
  count_col,
  max_radius_km = 3,
  outfile = NULL
)
```

Arguments

`polys`	data.frame with one row per polygon, including a `centroid_lat` / `centroid_lon` (or `LAT_WGS84` / `LONG_WGS84`) and the count column.
`count_col`	Name of the numeric column.
`max_radius_km`	Largest symbol radius in km.
`outfile`	Optional output path.

Value

ggplot object or invisible NULL.

`morie_tps_render_points`
Render a TPS point-pattern map (incident dots, optional DBSCAN)

Description

Projects (LAT_WGS84, LONG_WGS84) to the rotated Toronto canvas and draws one dot per incident. When `eps_km` and `min_samples` are supplied AND the **dbscan** package is installed, points are coloured by DBSCAN cluster label.

Usage

```
morie_tps_render_points(
  df,
  category = "Assault",
  eps_km = NULL,
  min_samples = 20L,
  outfile = NULL,
  show_top = 12L,
  fig_w = 12,
  fig_h = 7.5
)
```

Arguments

`df`	A TPS data.frame with columns `LAT_WGS84` and `LONG_WGS84`.
`category`	Optional category label used in the title.
`eps_km`	DBSCAN neighbourhood radius in km. When `NULL` no clustering is run.
`min_samples`	DBSCAN minimum cluster size.
`outfile`	Path to write the image, or `NULL` to return the plot.
`show_top`	Cap on how many clusters appear in the legend.
`fig_w`, `fig_h`	Figure size.

Value

A `ggplot` (when ggplot2 is available) or `invisible(NULL)` for the base-R path.

`morie_tps_render_quad` *Four-panel composite of TPS rendering primitives*

Description

Lays out a 2x2 quad combining a choropleth, point pattern, yearly grid summary, and (when available) a DBSCAN cluster panel. Falls back to base graphics with `par(mfrow = c(2, 2))` when ggplot2 is absent.

Usage

```
morie_tps_render_quad(data, outfile = NULL, ...)
```

Arguments

`data`	Named list with elements: `polys` (polygons frame), `points` (lat/lon points), `count_col`, `year_cols` (character vector of column names like `ASSAULT_RATE_2020:2024`).
`outfile`	Optional output path; when NULL the rendered object is returned (ggplot or invisible NULL for base).
`...`	Forwarded to the underlying single-panel renderers.

Value

A patchwork-or-list object (ggplot2 path) or invisible NULL.

`morie_tps_render_satscan_panel`
SaTScan-style spatial scan panel

Description

Renders Kulldorff-style circular candidate windows on the TPS canvas. Currently a thin layer over centroids + radius circles; the full likelihood-ratio overlay and significance ranking depend on the Python `morie.tps_satscan` module and are stubbed.

Usage

```
morie_tps_render_satscan_panel(clusters, outfile = NULL)
```

Arguments

`clusters`	data.frame with columns `lat` / `lon` / `radius_km` and optionally `llr` (log-likelihood ratio) for shading.
`outfile`	Optional output path.

Value

ggplot object or invisible NULL.

`morie_tps_render_yearly_grid`
Small-multiples panel of per-year TPS choropleths

Description

Walks `polys` once and renders one ggplot facet per year for columns named `<prefix>_<year>`.

Usage

```
morie_tps_render_yearly_grid(
  polys,
  prefix = "ASSAULT_RATE",
  years = 2014:2024,
  cmap = "Reds",
  outfile = NULL,
  ncols = 4L
)
```

Arguments

polys	Polygon data.frame (see morie_tps_render_choropleth).
prefix	Column-name prefix (default "ASSAULT_RATE").
years	Integer vector of years (default 2014:2024).
cmap	Sequential palette name (default "Reds").
outfile	Optional output path.
ncols	Number of facet columns.

Value

A ggplot (when ggplot2 is loaded) or invisible(NULL) for the base-R fallback.

---

morie_tps_resolve_hood_col

Resolve which HOOD_* column to use on a TPS crime data.frame

Description

Many TPS PSDP crime layers carry BOTH HOOD_158 (current) and HOOD_140 (historical 2014–2021) columns. Pick the version your analysis needs explicitly so the two schemes are not silently mixed across years.

Usage

```
morie_tps_resolve_hood_col(df, prefer = c("158", "140"), fallback = TRUE)
```

Arguments

df	A TPS crime data.frame.
prefer	Either "158" (current) or "140" (historical).
fallback	If TRUE (default), accept the other version when the preferred one is absent (with a warning). If FALSE, return NULL when the preferred version is missing.

Value

Character scalar (the chosen column name), or NULL if no suitable column is present.

Examples

```
df <- data.frame(OCC_YEAR = 2024L, HOOD_158 = "82", HOOD_140 = "82")
morie_tps_resolve_hood_col(df, prefer = "158")
```

`morie_tps_ripley_k` *Ripley's K function at multiple radii*

Description

Computes Ripley's K(r) at each user-supplied radius (km), the Besag-centred L(r)-r transformation, and the CSR baseline pi*r^2. Coordinates are projected to km via the small-angle latitude factor; for typical city-scale point patterns this is accurate enough that haversine is unnecessary.

Usage

```
morie_tps_ripley_k(
  df,
  ds_name = "?",
  radii_km = c(0.25, 0.5, 1, 2, 3, 5),
  max_n = 5000L,
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84"
)
```

Arguments

`df`	Incident-level data.frame.
`ds_name`	Tag for the result title.
`radii_km`	Numeric vector of radii in km (default 0.25, 0.5, 1, 2, 3, 5).
`max_n`	Subsample cap (default 5000) to keep the pairwise distance matrix tractable.
`lat_col`, `lon_col`	WGS84 column names.

Value

A named list with the per-radius table, intensity, and bounding-box area.

Examples

```
set.seed(2026)
df <- data.frame(
  LAT_WGS84 = 43.6 + rnorm(80, 0, 0.04),
  LONG_WGS84 = -79.4 + rnorm(80, 0, 0.04)
)
morie_tps_ripley_k(df, radii_km = c(0.5, 1, 2))
```

morie_tps_sarima_forecast

Seasonal ARIMA forecast on monthly incident counts

Description

Hold-out validation forecast: fits SARIMA(p,d,q)(P,D,Q)_12 with `stats::arima` to the leading training months, forecasts the last `h` months, and reports MAPE / RMSE.

Usage

```
morie_tps_sarima_forecast(
  df,
  ds_name = "?",
  h = 12L,
  order = c(1L, 1L, 1L),
  seasonal = c(0L, 1L, 1L, 12L)
)
```

Arguments

`df`	A `data.frame`.
`ds_name`	Character label.
`h`	Hold-out horizon in months (default 12).
`order`	Non-seasonal ARIMA order `c(p,d,q)`.
`seasonal`	Seasonal order `c(P,D,Q,s)`; the 4th element is the seasonal period.

Value

A `morie_rich_result` list with `aic`, `bic`, `mape_pct`, `rmse`, `forecast`, `actual`.

morie_tps_sdb_reaction_diffusion

Short-D'Orsogna-Brantingham 2008 hot-spot PDE

Description

Solves the coupled reaction-diffusion system

$$\partial_t A = \eta \nabla^2 A - \omega A + \theta \rho,$$

$$\partial_t \rho = \nabla \cdot (D \nabla \rho - 2\rho \nabla \log A) - \rho A + \gamma,$$

on a cosine-corrected Toronto grid seeded by the observed incident histogram. Localised attractiveness spikes emerge whenever $(\eta, \omega, \theta, D, \gamma)$ place the system in the instability regime (D'Orsogna & Perc 2015, sec. 3.2).

Usage

```
morie_tps_sdb_reaction_diffusion(
  category = "Assault",
  sample_rows = 30000L,
  eta = 0.05,
  omega = 0.3,
  theta = 1.5,
  D = 0.1,
  gamma = 0.05,
  n_steps = 800L,
  dt = 0.04,
  nx = 90L,
  ny = 60L,
  save_fig = TRUE
)
```

Arguments

`category`	TPS category name (default `"Assault"`).
`sample_rows`	Maximum number of incident rows to load (`NULL` for all).
`eta`, `omega`, `theta`, `D`, `gamma`	PDE coefficients.
`n_steps`	Number of forward-Euler integration steps.
`dt`	Integration step size.
`nx`, `ny`	Grid resolution.
`save_fig`	Whether to write a 1x3 PNG triptych (seed / A(x,t) / rho(x,t)) to the manifest figure directory.

Details

Steady-state spike count is compared against a DBSCAN cluster count on the raw incidents (delegated to `morie_tps_dbscan_clusters` when available).

Value

A `morie_rich_result` list with the steady-state spike count, mean field values, DBSCAN comparison, and the integration parameters.

References

Short MB, D'Orsogna MR, Pasour VB, Tita GE, Brantingham PJ, Bertozzi AL, Chayes LB (2008). A statistical model of criminal behavior. *M3AS* 18(supp01): 1249-1267.

Examples

```
## Not run:
  rr <- morie_tps_sdb_reaction_diffusion(
    "Assault", sample_rows = 5000, n_steps = 200, save_fig = FALSE
```

```
  )
  print(rr$summary_lines)

## End(Not run)
```

`morie_tps_sdb_turing_demo` *Canonical Short-D'Orsogna-Brantingham Turing-pattern demo*

Description

Reproduces the localised hot-spot lattice from Short, D'Orsogna & Brantingham (2008) Fig. 4 / D'Orsogna & Perc (2015) Fig. 5 on a clean periodic grid, seeded by a homogeneous steady state plus small Gaussian noise. The parameters chosen here place the system in the Turing-instability regime so the homogeneous solution is unstable and the system self-organises into a near-hexagonal lattice of localised spikes.

Usage

```
morie_tps_sdb_turing_demo(
  eta = 0.2,
  omega = 0.033,
  theta = 0.56,
  D = 30,
  gamma = 0.019,
  n_steps = 6000L,
  dt = 0.005,
  n = 80L,
  save_fig = TRUE
)
```

Arguments

`eta`, `omega`, `theta`, `D`, `gamma`	PDE coefficients.
`n_steps`	Integration steps.
`dt`	Step size.
`n`	Grid side length.
`save_fig`	Whether to write a 1x3 snapshot panel PNG.

Value

A `morie_rich_result` with the steady-state spike count, mean fields, and the integration parameters.

References

Short MB, D'Orsogna MR, Brantingham PJ *et al.* (2008). M3AS 18(supp01): 1249-1267.

Examples

```
## Not run:
  rr <- morie_tps_sdb_turing_demo(n = 32L, n_steps = 300L,
                                  save_fig = FALSE)
  print(rr$summary_lines$SteadySpikes)

## End(Not run)
```

---

morie_tps_seasonal_pattern

Seasonal / cyclic incident-time patterns

Description

Counts incidents by month-of-year, day-of-week, and hour-of-day, then runs a chi-square goodness-of-fit test against a uniform distribution on each cycle.

Usage

```
morie_tps_seasonal_pattern(df, ds_name = "?")
```

Arguments

`df`	A `data.frame`.
`ds_name`	Character label.

Value

A `morie_rich_result` list with per-cycle counts and chi-square p-values.

---

morie_tps_spatial_summary

Spatial rollup for a TPS crime data.frame

Description

Neighbourhood + division + premises + location-type rollups plus a lat/long bounding-box summary. Tolerates missing columns.

Usage

```
morie_tps_spatial_summary(df, ds_name = "?")
```

Arguments

`df`	A TPS crime data.frame.
`ds_name`	Optional dataset label used in the result title.

Value

A `morie_tps_result` named list.

`morie_tps_statphysics`

Statistical physics of crime for TPS data

Description

R port of `morie.tps_statphysics`. Implements the four canonical methods reviewed by D'Orsogna & Perc (2015), *Statistical physics of crime: A review*, Physics of Life Reviews 12: 1-21 (arXiv:1411.1743), together with two illustrative companions (canonical Turing-pattern demo and Helbing-Szolnoki inspection-game phase diagram) and a premise x neighbourhood co-occurrence network.

Details

Each callable consumes one TPS category and returns a multi-section `morie_rich_result`. Cosine-corrected projection and DBSCAN delegation are deferred to companion modules (`tps_render`, `tps_spatial_advanced`); when those collaborators are not available the routines fall back to a stop-stub explaining the gap.

Functions:

- `morie_tps_sdb_reaction_diffusion` — Short, D'Orsogna and Brantingham (2008) hot-spot PDE, data-seeded.
- `morie_tps_levy_flight_alpha` — Hill-MLE Levy-flight tail exponent following Brockmann, Hufnagel and Geisel (2006).
- `morie_tps_urban_scaling_beta` — Bettencourt *et al.* (2007) urban-scaling beta across the 158 Toronto wards.
- `morie_tps_lotka_volterra_police_crime` — Lotka- Volterra predator-prey on yearly counts.
- `morie_tps_sdb_turing_demo` — canonical Turing- instability demo on a periodic lattice.
- `morie_tps_inspection_game_phase` — three- strategy replicator phase diagram (Helbing, Szolnoki & Perc 2010).
- `morie_tps_criminal_network_graph` — premise x neighbourhood co-occurrence network (Diviak *et al.* 2019-style projection from public TPS data).

References

D'Orsogna MR, Perc M (2015). Statistical physics of crime: A review. *Physics of Life Reviews* 12: 1-21.

Short MB, D'Orsogna MR, Pasour VB, Tita GE, Brantingham PJ, Bertozzi AL, Chayes LB (2008). A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences* 18(supp01): 1249-1267.

Brockmann D, Hufnagel L, Geisel T (2006). The scaling laws of human travel. *Nature* 439: 462-465.

Bettencourt LMA, Lobo J, Helbing D, Kuhnert C, West GB (2007). Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the National Academy of Sciences* 104: 7301-7306.

Helbing D, Szolnoki A, Perc M, Szabo G (2010). Punish, but not too hard: how costly punishment spreads in the spatial public goods game. *New Journal of Physics* 12: 083005.

Diviak T, Dijkstra JK, Snijders TAB (2019). Structure, multiplexity, and centrality in a corruption network. *Trends in Organized Crime* 22: 274-297.

`morie_tps_statphysics_analyze_all`
Run all four statistical-physics analyses on a list of categories

---

Description

Convenience wrapper that calls `morie_tps_sdb_reaction_diffusion`, `morie_tps_levy_flight_alpha`, `morie_tps_urban_scaling_beta`, and `morie_tps_lotka_volterra_police_crime` on every category in the supplied list. Returns a nested list keyed first by category and then by method.

Usage

```
morie_tps_statphysics_analyze_all(categories = NULL, save_fig = TRUE)
```

Arguments

`categories`	Character vector of TPS category names; default is the canonical nine-category TPS set.
`save_fig`	Whether to ask each sub-routine to write its figure.

Value

A named list of lists of `morie_rich_result` objects.

References

D'Orsogna MR, Perc M (2015). *Physics of Life Reviews* 12: 1-21.

Examples

```
## Not run:
  res <- morie_tps_statphysics_analyze_all(c("Assault", "Robbery"),
                                           save_fig = FALSE)

## End(Not run)
```

---

morie_tps_stochastic *Stochastic-physics-of-crime analyses for TPS data*

Description

R port of `morie.tps_stochastic`. Four jurisdiction-agnostic callables: temporal-only exponential Hawkes self-exciting fit, seasonal ARIMA forecast on monthly counts, Euler-Maruyama Ornstein-Uhlenbeck simulation, and a 1-D Fokker-Planck density evolution. The R port keeps optimisation in base R (`stats::optim`) and the seasonal forecast in `stats::arima` so no external time-series package is needed.

Details

All functions return a multi-section `morie_rich_result` list.

Functions:

- `morie_tps_hawkes_temporal_fit`: fit mu, kappa, omega of an exponential-kernel Hawkes process to incident times; report branching ratio + AIC/BIC.
- `morie_tps_sarima_forecast`: seasonal ARIMA on monthly counts with train / hold-out MAPE.
- `morie_tps_langevin_simulate`: Euler-Maruyama OU SDE paths fitted to daily counts.
- `morie_tps_fokker_planck_grid`: 1-D finite-difference density evolution under OU drift+diffusion.

References: Mohler et al. 2011 (self-exciting point process crime); Short, D'Orsogna, Bertozzi 2010 (stochastic physics of crime).

MORIE_TPS_SUPPORTED_FORMATS

Format names that `morie_tps_load()` *knows how to dispatch.*

Description

Format names that `morie_tps_load()` knows how to dispatch.

Usage

```
MORIE_TPS_SUPPORTED_FORMATS
```

Format

An object of class `character` of length 9.

---

morie_tps_temporal *Temporal analyses for TPS crime data*

Description

R port of `morie.tps_temporal`. Four jurisdiction-agnostic callables operating on a Toronto Police Service-shaped `data.frame`: yearly trend, seasonal cyclic stats, Pettitt-style change-point on yearly counts, and an ARIMA(1,1,1) forecast on monthly counts. All functions return a multi-section `morie_rich_result` list so output can be printed directly to a notebook.

Details

Functions:

- `morie_tps_year_over_year_trend`: OLS slope / intercept / R-squared on yearly incident counts.
- `morie_tps_seasonal_pattern`: month / DOW / hour cyclic counts plus chi-square uniformity tests.
- `morie_tps_changepoint_detection`: Pettitt's non-parametric change-point on yearly counts (no external change-point dependency).
- `morie_tps_arima_forecast`: ARIMA(1,1,1) forecast on monthly counts via `stats::arima`.

---

morie_tps_temporal_summary *Temporal rollup for a TPS crime data.frame*

Description

Year / month / day-of-week / hour-of-day rollups, plus a coverage line. Robust to missing columns: only includes tables for the fields actually present.

Usage

```
morie_tps_temporal_summary(df, ds_name = "?")
```

Arguments

`df`	A TPS crime data.frame.
`ds_name`	Optional dataset label used in the result title.

Value

A `morie_tps_result` named list.

`MORIE_TPS_TORONTO_POPULATION_BY_YEAR`
Toronto reference population by fiscal year (StatsCan 17-10-0009-01).

Description

Toronto reference population by fiscal year (StatsCan 17-10-0009-01).

Usage

```
MORIE_TPS_TORONTO_POPULATION_BY_YEAR()
```

Value

Named integer vector (year-as-string -> population).

`MORIE_TPS_TOTAL_CSI_WEIGHTS`
Total-CSI weights for the 9 TPS open-data categories.

Description

Total-CSI weights for the 9 TPS open-data categories.

Usage

```
MORIE_TPS_TOTAL_CSI_WEIGHTS()
```

Value

Named numeric vector.

`morie_tps_urban_scaling_beta` *Bettencourt urban-scaling exponent across the 158 Toronto wards*

Description

Performs the standard log-log OLS scaling fit

$$\log y_i = \log Y_0 + \beta \log p_i + \varepsilon_i,$$

where y_i is the crime count and p_i is the population of ward `i`. $\beta > 1$ indicates super-linear (crime grows faster than population), $\beta = 1$ linear, and $\beta < 1$ sub-linear (protective) scaling (Bettencourt *et al.* 2007; D'Orsogna & Perc 2015 sec. 4.1).

Usage

```
morie_tps_urban_scaling_beta(
  category = "Assault",
  year = 2024L,
  save_fig = TRUE
)
```

Arguments

`category`	TPS category name.
`year`	Reference year used to choose the appropriate population and crime columns.
`save_fig`	Whether to write a log-log scatter + fit PNG.

Value

A `morie_rich_result` with $\hat{\beta}$, its standard error, R-squared, the back-transformed prefactor Y_0, and a regime label (sub-linear, linear, super-linear).

References

Bettencourt LMA, Lobo J, Helbing D, Kuhnert C, West GB (2007). Growth, innovation, scaling, and the pace of life in cities. *PNAS* 104: 7301-7306.

Examples

```
## Not run:
  rr <- morie_tps_urban_scaling_beta("Assault", year = 2024,
                                     save_fig = FALSE)
  print(rr$summary_lines)

## End(Not run)
```

`morie_tps_use_of_force`
TPS Use-of-Force rate + type distribution

Description

Compact callable mirroring `morie.fn.tpsuof.tps_use_of_force`. Computes a use-of-force rate over a known encounter denominator and returns a per-type count distribution, packaged as a rich-result list compatible with the `morie_mrm_uof_result` family in `R/mrm_uof.R`.

Usage

```
morie_tps_use_of_force(force_types, n_encounters)

morie_tpsuof(force_types, n_encounters)
```

Arguments

`force_types`	Character vector of use-of-force-type labels (one row per use-of-force incident).
`n_encounters`	Positive integer total number of police-public encounters in the denominator.

Details

Formula:

- `rate = length(force_types) / n_encounters`
- `type_counts = table(force_types)`

Value

A named `list` with classes `morie_tps_use_of_force_result`, `morie_mrm_uof_result`, `morie_rich_result`, `list`. Slots: `rate`, `n`, `population`, `type_counts`, `n_types`, `interpretation`.

Examples

```
force_types <- c("Physical Control", "Physical Control", "CEW",
                 "Firearm", "OC Spray")
morie_tps_use_of_force(force_types, n_encounters = 1000L)
```

`MORIE_TPS_VIOLENT_CSI_WEIGHTS`
Violent-CSI weights for the 9 TPS open-data categories.

Description

Violent-CSI weights for the 9 TPS open-data categories.

Usage

```
MORIE_TPS_VIOLENT_CSI_WEIGHTS()
```

Value

Named numeric vector.

`morie_tps_year_over_year_trend`
Year-over-year linear trend on incident counts

Description

Aggregates incident counts by year, restricts to the 1990-2030 window, fits an OLS line, and reports slope, intercept, and R-squared.

Usage

```
morie_tps_year_over_year_trend(df, year_col = "OCC_YEAR", ds_name = "?")
```

Arguments

`df`	A `data.frame` with one row per incident.
`year_col`	Character. Name of the year column (default `"OCC_YEAR"`).
`ds_name`	Character label for the dataset shown in titles.

Value

A `morie_rich_result` list with `slope`, `intercept`, `r2`, `direction`, `years`, `counts`, `fitted`.

morie_tps_year_to_hood_version

Recommended HOOD_* schema version for a given OCC_YEAR

Description

The City of Toronto adopted the 158-neighbourhood scheme in 2022. Pre-2022 TPS crime records are most faithfully analysed in the historical 140-neighbourhood scheme; 2022-onwards records align with the 158-scheme. TPS often back-fills both columns onto the same record via lat/lon re-geocoding, but the polygon boundaries do not match.

Usage

```
morie_tps_year_to_hood_version(year)
```

Arguments

`year`	Integer year (or vector of years).

Value

Character vector of `"158"` / `"140"` recommendations, parallel to `year`.

morie_tps_yoy_panel

Side-by-side year-over-year panel across TPS categories

Description

For each named TPS category in `dfs`, groups by the dataset's year column (`OCC_YEAR` preferred, `REPORT_YEAR` fallback), restricts to plausible years (1990-2030), and joins all series column-wise into a single panel of incident counts.

Usage

```
morie_tps_yoy_panel(dfs, categories = NULL)
```

Arguments

`dfs`	Named list of TPS data.frames keyed by category name.
`categories`	Optional character vector restricting which keys of `dfs` are analysed; defaults to all of them.

Value

A `morie_tps_result` named list with a single year-by-category table.

`morie_transformer_genomic` *Transformer (1-head self-attention) genomic predictor (base R)*

Description

Random fixed projections + ridge head on the mean-pooled context vector.

Usage

```
morie_transformer_genomic(
  x,
  y,
  markers,
  d_model = 8,
  lam = 1,
  seed = 0,
  deterministic_seed = NULL
)
```

Arguments

`x`	Optional fixed-effect features.
`y`	Numeric response.
`markers`	(n x L) marker sequence.
`d_model`	Model dimension.
`lam`	Ridge regulariser for the linear head.
`seed`	Seed.
`deterministic_seed`	Optional integer; if supplied, RNG state is derived via `morie_det_rng()` keyed on ("trfge", deterministic_seed) so Py<->R streams agree on the canonical fixture. When `NULL` (default) behaviour is unchanged.

Value

list(estimate, y_hat, beta, attention, context, se, n, method).

References

Vaswani et al. (2017). Montesinos Lopez Ch 15.

Examples

```
morie_transformer_genomic(
  x = rnorm(50), y = rnorm(50),
  markers = matrix(sample(0:2, 200, TRUE), 50, 4)
)
```

`morie_tsne_reduction` *t-SNE for non-linear dimension reduction (R parity)*

Description

Wraps `Rtsne::Rtsne`.

Usage

```
morie_tsne_reduction(
  x,
  n_components = 2L,
  perplexity = 30,
  learning_rate = "auto",
  n_iter = 1000L,
  seed = 0L,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric matrix.
`n_components`	Embedding dimension.
`perplexity`	t-SNE perplexity.
`learning_rate`	Unused by Rtsne (kept for API parity).
`n_iter`	Max iterations.
`seed`	RNG seed.
`deterministic_seed`	Integer or NULL. If supplied, the RNG state is derived from the SHA-keyed `morie_det_rng()` so Py<->R streams agree on the canonical fixture. When `NULL` (default), behaviour is unchanged: `seed` drives `set.seed()` directly.

Value

Named list: estimate (shape), embedding, kl_divergence, perplexity, n_components, n, method.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

morie_two_sample_coverage

Two-sample placement coverage (Gibbons Ch 2.11.2)

Description

Block frequencies of Y in the m+1 intervals defined by the ordered X-sample. Under H0 the expected block proportion is `1 / (m + 1)`.

Usage

```
morie_two_sample_coverage(x, y)
```

Arguments

`x`	Numeric vector (first sample).
`y`	Numeric vector (second sample).

Value

Named list: block_freq, block_prop, expected_prop, m, n, cumulative, method.

Examples

```
morie_two_sample_coverage(x = rnorm(50), y = rnorm(50))
```

morie_two_sample_t_test

Two-sample t-test with tidy output

Description

Two-sample t-test with tidy output

Usage

```
morie_two_sample_t_test(
  x1,
  x2,
  equal_var = FALSE,
  alternative = c("two.sided", "greater", "less")
)
```

Arguments

`x1`	Numeric vector (group 1).
`x2`	Numeric vector (group 2).
`equal_var`	Assume equal variances? Default `FALSE` (Welch test).
`alternative`	`"two.sided"`, `"greater"`, or `"less"`.

Value

Named list: `t`, `df`, `p_value`, `ci_diff`, `cohens_d`.

Examples

```
morie_two_sample_t_test(rnorm(50, 0.5), rnorm(50, 0))
```

---

morie_unobserved_components

Unobserved-components decomposition (trend + seasonal + irregular)

Description

Unobserved-components decomposition (trend + seasonal + irregular)

Usage

```
morie_unobserved_components(x, period = 12, trend = "local linear")
```

Arguments

`x`	Numeric univariate series.
`period`	Seasonal period (pass 0 to omit). Default 12.
`trend`	Trend specification, "local level" or "local linear".

Value

Named list with `trend, seasonal, irregular, loglik, n, period, method`.

Examples

```
morie_unobserved_components(x = rnorm(50))
```

morie_userguide *Get path to an MORIE userguide*

Description

Lists or retrieves bundled userguide PDF files. These are the official PUMF codebooks and user guides from Health Canada / Statistics Canada.

Usage

```
morie_userguide(name = NULL)
```

Arguments

name	Filename (e.g., "20212022-cpads-pumf-user-guide.pdf"). If NULL, lists all available userguides.

Value

File path string, or character vector of filenames.

Examples

```
morie_userguide()
```

morie_validate_cpads_data *Validate a CPADS analysis data frame*

Description

Validate a CPADS analysis data frame

Usage

```
morie_validate_cpads_data(data, strict = TRUE)
```

Arguments

data	Data frame to validate.
strict	If TRUE, stop when required variables are missing.

Value

Character vector of missing variable names.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_validate_outputs_manifest`
Validate outputs manifest structure

Description

Validate outputs manifest structure

Usage

```
morie_validate_outputs_manifest(manifest, strict = TRUE)
```

Arguments

`manifest`	Data frame to validate.
`strict`	If `TRUE`, stop on validation failures.

Value

`TRUE` when validation passes.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_van_der_waerden_test`
Van der Waerden two-sample normal-scores location test (Gibbons Ch 8.3.2)

Description

Scores a_i = qnorm(R_i / (N + 1)); statistic = sum over the first sample.

Usage

```
morie_van_der_waerden_test(x, y)
```

Arguments

`x`, `y`	Numeric vectors.

Value

Named list: statistic, p_value, z, n, m.

Examples

```
morie_van_der_waerden_test(x = rnorm(50), y = rnorm(50))
```

---

morie_vecm *Vector error-correction model (VECM)*

Description

Vector error-correction model (VECM)

Usage

```
morie_vecm(Y, k_ar = 1, coint_rank = 1)
```

Arguments

`Y`	Numeric matrix (T x k) of I(1) candidate series.
`k_ar`	Number of lagged differences. Default 1.
`coint_rank`	Cointegration rank. Default 1.

Value

Named list with `alpha`, `beta`, `Gamma`, `Sigma`, `loglik`, `n`, `k`, `rank`, `method`.

Examples

```
morie_vecm(Y = matrix(rnorm(100), 50, 2))
```

---

morie_verify_statistical_output *Verify that a serialised statistical output meets minimum quality gates*

Description

Mirrors the Python `morie.verify_statistical_output()`. Runs a small suite of sanity checks on a JSON output containing fields commonly used across MORIE estimators: `ate`, `se`, `ci_lower`, `ci_upper`, `n`, `p_value`. Each check is a named boolean; the verification passes if all checks are `TRUE`.

Usage

```
morie_verify_statistical_output(path)
```

Arguments

`path`	Path to a JSON output file.

Details

Checks: SE non-negative; CI lower < CI upper; estimate inside the CI; n positive; p-value (if present) in [0, 1]; estimate finite.

Value

A list with `path`, `passed` (logical), and `checks` (named list of boolean check results).

Examples

```
tmp <- tempfile(fileext = ".json")
if (requireNamespace("jsonlite", quietly = TRUE)) {
  jsonlite::write_json(
    list(ate = 0.5, se = 0.1, ci_lower = 0.3, ci_upper = 0.7, n = 200),
    tmp,
    auto_unbox = TRUE
  )
  morie_verify_statistical_output(tmp)
  unlink(tmp)
}
```

`morie_vertex_access_token`

Fetch and cache a Google Cloud access token via gcloud

Description

Fetch and cache a Google Cloud access token via gcloud

Usage

```
morie_vertex_access_token(cfg = NULL)
```

Arguments

`cfg`	Config list, or NULL to resolve.

Value

Character bearer token.

`morie_vertex_ask_gemini` *Send a single-turn prompt to Gemini via Vertex AI*

Description

R port of `morie.vertex.ask_gemini`. POSTs to the Vertex AI REST endpoint `:generateContent` and returns the concatenated text from the first candidate.

Usage

```
morie_vertex_ask_gemini(
  prompt,
  model = NULL,
  system = NULL,
  temperature = 0.1,
  max_output_tokens = 2048L,
  timeout_s = 120,
  cfg = NULL
)
```

Arguments

`prompt`	Character scalar – the user prompt.
`model`	Optional Gemini model override.
`system`	Optional system instruction.
`temperature`	Numeric. Default 0.1.
`max_output_tokens`	Integer. Default 2048.
`timeout_s`	Numeric HTTP timeout. Default 120.
`cfg`	Pre-resolved config list, or NULL to auto-resolve.

Value

Character scalar – trimmed generated text.

`morie_vertex_health_check`
Tiny smoke test for the Vertex AI client

Description

Tiny smoke test for the Vertex AI client

Usage

```
morie_vertex_health_check()
```

Value

Named list (ok / error / model / project / location / reply).

`morie_vertex_resolve_config`
Resolve Vertex AI configuration from environment variables

Description

Resolve Vertex AI configuration from environment variables

Usage

```
morie_vertex_resolve_config()
```

Value

Named list: project / location / model / token_ttl_s / gcloud_path.

`morie_vpd_download_instructions`
Print step-by-step VPD GeoDASH download instructions

Description

VPD's GeoDASH Open Data portal (https://geodash.vpd.ca/opendata/) has no automation API: every download requires a manual click- through of VPD's terms-of-use plus a popup-based file save per (year, neighbourhood) selection. Calling `morie_vpd_download_instructions()` prints the exact steps the user needs to follow to get a full-fidelity crime CSV that `morie_datasets_vpd_crime` can consume.

Usage

```
morie_vpd_download_instructions(to = NULL)
```

Arguments

to	Optional file path. If supplied, the instructions are ALSO written to that file (so the user can read them outside R). Default NULL = print to console only.

Value

Invisibly the instruction text (character vector, one line per element).

See Also

morie_datasets_vpd_crime() for the loader that accepts the downloaded file.

Examples

```
morie_vpd_download_instructions()
```

---

morie_wavelet_time_series

Discrete wavelet decomposition for a time series

Description

Discrete wavelet decomposition for a time series

Usage

```
morie_wavelet_time_series(x, wavelet = "haar", level = NULL)
```

Arguments

x	Numeric univariate series.
wavelet	Wavelet family. Default "haar".
level	Decomposition depth. Default floor(log2 n) capped at 6.

Value

Named list with approximation, details, energies, level, n, wavelet, method.

Examples

```
morie_wavelet_time_series(x = rnorm(50))
```

morie_weights_bootstrap

Bootstrap replicate weights (Rao-Wu rescaling within strata).

Description

Bootstrap replicate weights (Rao-Wu rescaling within strata).

Usage

```
morie_weights_bootstrap(weights, n_replicates = 200, strata = NULL, seed = 42)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`n_replicates`	Integer; replicate count.
`strata`	Optional vector of stratum identifiers aligned with `weights`.
`seed`	Integer RNG seed.

morie_weights_brr

Balanced Repeated Replication (BRR) weights.

Description

Each stratum is split into two halves; signs from a random Hadamard-like matrix double one half and zero the other. For exact Hadamard ordering use `survey::as.svrepdesign(..., type = "BRR")`.

Usage

```
morie_weights_brr(weights, strata, n_replicates = NULL, seed = 42)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`strata`	Optional vector of stratum identifiers aligned with `weights`.
`n_replicates`	Integer; replicate count.
`seed`	Integer RNG seed.

`morie_weights_calibrate_to_totals`
Dispatch helper – calibrate to totals via "raking" or "greg".

Description

Dispatch helper – calibrate to totals via "raking" or "greg".

Usage

```
morie_weights_calibrate_to_totals(
  weights,
  df,
  totals,
  method = c("raking", "greg"),
  ...
)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`df`	A `data.frame` of unit-level covariates aligned with `weights`.
`totals`	Named numeric vector of calibration targets used by `calibrate_to_totals`.
`method`	Character; calibration / smoothing / variance method. Allowed values depend on the caller.
`...`	Additional method-specific arguments.

`morie_weights_combined`
Combined design x nonresponse x post-strat (x trim) pipeline.

Description

Combined design x nonresponse x post-strat (x trim) pipeline.

Usage

```
morie_weights_combined(
  selection_probs,
  responded,
  adjustment_cells = NULL,
  calibration_strata = NULL,
  population_totals = NULL,
  trim_percentiles = NULL
)
```

Arguments

`selection_probs`
: Numeric vector of selection probabilities for `design`.

`responded`
: Logical/integer 0/1 vector of response indicators.

`adjustment_cells`
: Optional cell identifiers used by `nonresponse` adjustment.

`calibration_strata`
: Optional strata for the calibration step of `combined`.

`population_totals`
: Named numeric vector of target totals to calibrate to.

`trim_percentiles`
: Two-element numeric vector `c(lower, upper)` of percentile bounds for trimming.

`morie_weights_deff` *Kish design effect (n / ESS).*

Description

Kish design effect (n / ESS).

Usage

```
morie_weights_deff(weights)
```

Arguments

`weights`
: Numeric vector of unit-level design weights.

`morie_weights_design` *Design weights from inclusion probabilities.*

Description

$w_i = 1/\pi_i$.

Usage

```
morie_weights_design(selection_probs)
```

Arguments

`selection_probs`
: Numeric vector of selection probabilities for `design`.

`morie_weights_detect_extreme`
*Detect extreme weights at +/- k * IQR or by absolute percentile.*

Description

Detect extreme weights at +/- k * IQR or by absolute percentile.

Usage

```
morie_weights_detect_extreme(weights, k = 3)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`k`	Numeric multiplier for `detect_extreme` (Tukey MAD cut).

`morie_weights_diagnostics`
Comprehensive weight diagnostics.

Description

Returns a named list with summary statistics, Kish ESS, design effect, weight-range ratio, and percentile vector.

Usage

```
morie_weights_diagnostics(weights)
```

Arguments

`weights`	Numeric vector of unit-level design weights.

`morie_weights_ess` *Kish effective sample size:* $(\sum w_i)\hat{}2/\sum w_i\hat{}2$.

Description

Kish effective sample size: $(\sum w_i)^2 / \sum w_i^2$.

Usage

```
morie_weights_ess(weights)
```

Arguments

`weights`	Numeric vector of unit-level design weights.

`morie_weights_fay_brr` *Fay's BRR weights with perturbation coefficient* `fay_coefficient` *in [0,1).*

Description

Fay's BRR weights with perturbation coefficient `fay_coefficient` in [0,1).

Usage

```
morie_weights_fay_brr(
  weights,
  strata,
  fay_coefficient = 0.5,
  n_replicates = NULL,
  seed = 42
)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`strata`	Optional vector of stratum identifiers aligned with `weights`.
`fay_coefficient`	Fay's coefficient (k) for Fay's-BRR.
`n_replicates`	Integer; replicate count.
`seed`	Integer RNG seed.

`morie_weights_greg` *Generalised regression (GREG) calibration.*

Description

Closed-form linear calibration to match population totals on auxiliary X. When `survey` is installed, defers to `survey::calibrate()` for a fully design-aware result; otherwise computes the linear adjustment in base R.

Usage

```
morie_weights_greg(weights, X, population_totals, max_iter = 50, tol = 1e-08)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`X`	A numeric matrix or `data.frame` of auxiliary variables (one row per unit) used by GREG / nonresponse propensity models.
`population_totals`	Named numeric vector of target totals to calibrate to.
`max_iter`	Iteration cap for calibration / IPF.
`tol`	Convergence tolerance for calibration.

`morie_weights_jackknife` *Jackknife replicate weights (JK1 delete-1 or JKn stratified delete-n).*

Description

When the `survey` package is installed and `strata` is supplied, defers to `survey::as.svrepdesign(..., type = "JKn")` for variance compatibility.

Usage

```
morie_weights_jackknife(weights, strata = NULL, jk_type = c("JK1", "JKn"))
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`strata`	Optional vector of stratum identifiers aligned with `weights`.
`jk_type`	Character; jackknife type (`"jk1"`, `"jk2"`, `"jkn"`).

morie_weights_multiframe

Multi-frame (dual-frame) survey weights (Hartley compositing).

Description

Multi-frame (dual-frame) survey weights (Hartley compositing).

Usage

```
morie_weights_multiframe(
  weights_a,
  weights_b,
  overlap_a,
  overlap_b,
  method = c("hartley", "optimal"),
  theta = 0.5
)
```

Arguments

`weights_a`	Numeric vector of weights from frame A (`multiframe`).
`weights_b`	Numeric vector of weights from frame B (`multiframe`).
`overlap_a`	Logical vector flagging frame-A units in overlap.
`overlap_b`	Logical vector flagging frame-B units in overlap.
`method`	Character; calibration / smoothing / variance method. Allowed values depend on the caller.
`theta`	Numeric tuning parameter passed to `multiframe` (Hartley-style overlap composition).

morie_weights_nonresponse

Non-response adjustment within cells.

Description

Within each cell, scales respondent weights up by total/responder ratio. Non-respondents end up with weight 0.

Usage

```
morie_weights_nonresponse(weights, responded, adjustment_cells = NULL)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`responded`	Logical/integer 0/1 vector of response indicators.
`adjustment_cells`	Optional cell identifiers used by `nonresponse` adjustment.

`morie_weights_normalize` *Normalise weights so they sum to n (sample) or N (population).*

Description

Normalise weights so they sum to n (sample) or N (population).

Usage

```
morie_weights_normalize(
  weights,
  target = c("sample_size", "population"),
  population_size = NULL
)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`target`	Numeric target sum (`normalize`).
`population_size`	Optional known population size used by `normalize` (target sum = population_size when supplied).

`morie_weights_poststratify` *Post-stratification weight adjustment.*

Description

$w_i^{ps} = w_i \cdot N_h / \hat{N}_h$.

Usage

```
morie_weights_poststratify(weights, strata, population_totals)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`strata`	Optional vector of stratum identifiers aligned with `weights`.
`population_totals`	Named numeric vector of target totals to calibrate to.

`morie_weights_propensity_nonresponse` *Propensity-score non-response weights (logistic).*

Description

Propensity-score non-response weights (logistic).

Usage

```
morie_weights_propensity_nonresponse(weights, responded, X)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`responded`	Logical/integer 0/1 vector of response indicators.
`X`	A numeric matrix or `data.frame` of auxiliary variables (one row per unit) used by GREG / nonresponse propensity models.

`morie_weights_rake` *Raking calibration (iterative proportional fitting).*

Description

Adjusts weights so that within each calibration variable the weighted sums match the supplied marginal targets. `margins` is a named list keyed by variable name; each entry is a named numeric vector mapping category values (as strings) to target totals.

Usage

```
morie_weights_rake(
  weights,
  df,
  margins,
  max_iter = 100,
  tol = 1e-06,
  bounds = NULL
)
```

Arguments

`weights`	Initial numeric weights (length n).
`df`	data.frame containing the calibration variables.
`margins`	Named list of named numeric vectors.
`max_iter`	Maximum IPF iterations (default 100).
`tol`	Convergence tolerance on max relative adjustment (default 1e-6).
`bounds`	Optional `c(lo, hi)` to clip the per-iteration multiplier.

Value

list with `weights`, `converged`, `iterations`, `max_adjustment`, `diagnostics` (from `morie_weights_diagnostics`).

`morie_weights_replicate_variance` *Variance estimation from replicate estimates.*

Description

`method` selects the rescaling: "JK1", "JKn", "BRR", "Fay", "bootstrap", "SDR".

Usage

```
morie_weights_replicate_variance(
  full_estimate,
  replicate_estimates,
  method = c("JK1", "JKn", "BRR", "Fay", "bootstrap", "SDR"),
  fay_coefficient = 0,
  strata = NULL
)
```

Arguments

`full_estimate`	Numeric scalar; full-sample point estimate.
`replicate_estimates`	Numeric vector of replicate point estimates.
`method`	Character; calibration / smoothing / variance method. Allowed values depend on the caller.
`fay_coefficient`	Fay's coefficient (k) for Fay's-BRR.
`strata`	Optional vector of stratum identifiers aligned with `weights`.

morie_weights_sdr *Successive Difference Replication (SDR) weights.*

Description

Successive Difference Replication (SDR) weights.

Usage

```
morie_weights_sdr(weights, n_replicates = 100, seed = 42)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`n_replicates`	Integer; replicate count.
`seed`	Integer RNG seed.

morie_weights_smooth *Smooth survey weights via shrinkage toward the mean (or log-mean).*

Description

Smooth survey weights via shrinkage toward the mean (or log-mean).

Usage

```
morie_weights_smooth(
  weights,
  method = c("linear_shrinkage", "log_transform"),
  shrinkage_factor = 0.5
)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`method`	Character; calibration / smoothing / variance method. Allowed values depend on the caller.
`shrinkage_factor`	Numeric in `[0, 1]`; smoothing pull-toward- mean factor.

morie_weights_trim *Trim extreme weights at percentile cutpoints.*

Description

`method = "percentile"` clips at the specified percentiles; `method = "winsorize"` replaces outliers with the boundary values.

Usage

```
morie_weights_trim(
  weights,
  lower_percentile = 1,
  upper_percentile = 99,
  method = c("percentile", "winsorize")
)
```

Arguments

`weights`	Numeric vector of unit-level design weights.
`lower_percentile`	Lower percentile cut for `trim`.
`upper_percentile`	Upper percentile cut for `trim`.
`method`	Character; calibration / smoothing / variance method. Allowed values depend on the caller.

morie_wilcoxon_power *Monte-Carlo power of the Wilcoxon signed-rank test (Gibbons Ch 5.7.3)*

Description

Simulates samples of size `length(x)` from Normal(effect_size, 1) and reports the rejection rate of two-sided wilcox.test at level alpha.

Usage

```
morie_wilcoxon_power(x, effect_size = 0.5, alpha = 0.05, nsim = 2000, seed = 0)
```

Arguments

`x`	Numeric vector (only `length(x)` is used).
`effect_size`	Location shift under H1.
`alpha`	Test level.
`nsim`	Replicates.
`seed`	Reproducibility seed (NULL = no fix).

Value

Named list: statistic (power), n, effect_size, alpha, nsim, se.

Examples

```
morie_wilcoxon_power(x = rnorm(50))
```

---

morie_wilcoxon_signed_rank_test

Wilcoxon signed-rank test (paired)

Description

Wilcoxon signed-rank test (paired)

Usage

```
morie_wilcoxon_signed_rank_test(
  x1,
  x2,
  alternative = c("two.sided", "greater", "less")
)
```

Arguments

`x1`	Numeric vector (before).
`x2`	Numeric vector (after).
`alternative`	`"two.sided"`, `"greater"`, or `"less"`.

Value

Named list: `V`, `p_value`.

Examples

```
# See the package vignettes for usage examples:
#   vignette(package = "rmorie")
```

`morie_write_audit_markdown` *Write a Markdown audit report.*

Description

Write a Markdown audit report.

Usage

```
morie_write_audit_markdown(out_path, audit_result)
```

Arguments

`out_path`	Path to write to.
`audit_result`	A `morie_audit_result` or list of them.

Value

The path written.

`morie_write_synthetic_data` *Write synthetic epidemiology-style data to CSV*

Description

Write synthetic epidemiology-style data to CSV

Usage

```
morie_write_synthetic_data(
  path,
  n = 5000L,
  seed = 42L,
  special_code_rate = 0.02,
  profile = c("generic", "morie_legacy"),
  name_map = NULL,
  overwrite = FALSE
)
```

Arguments

`path`	Output CSV path.
`n`	Number of rows.
`seed`	Random seed.
`special_code_rate`	Proportion of survey-style missing codes.
`profile`	Naming profile for output columns.
`name_map`	Optional custom variable name map.
`overwrite`	If `TRUE`, overwrite existing file.

Value

Normalized output path.

Examples

```
out <- morie_write_synthetic_data(tempfile(fileext = ".csv"), n = 200, seed = 1)
file.exists(out)
```

`morie_xgboost_objective` *XGBoost regularized objective (R parity)*

Description

Wraps the `xgboost` package. If xgboost isn't installed, falls back to `gbm` (gradient boosting) so users still get a usable boosted-trees result; the backend is flagged in the output.

Usage

```
morie_xgboost_objective(
  x,
  y,
  n_estimators = 100L,
  learning_rate = 0.1,
  max_depth = 3L,
  reg_lambda = 1,
  reg_alpha = 0,
  task = "auto",
  seed = 0L,
  deterministic_seed = NULL
)
```

Arguments

`x`	Numeric predictor matrix.
`y`	Response.
`n_estimators`	Number of boosting rounds.
`learning_rate`	eta / shrinkage.
`max_depth`	Tree depth.
`reg_lambda`	L2 leaf penalty.
`reg_alpha`	L1 leaf penalty.
`task`	"auto", "classification", or "regression".
`seed`	RNG seed.
`deterministic_seed`	Integer or NULL. If supplied, the RNG state is derived from the SHA-keyed `morie_det_rng()` so Py<->R streams agree on the canonical fixture. When `NULL` (default), behaviour is unchanged: `seed` drives `set.seed()` directly.

Value

Named list: estimate, train_score, feature_importances, backend, n_estimators, learning_rate, max_depth, reg_lambda, reg_alpha, task, n, method.

Examples

```
morie_xgboost_objective(x = rnorm(50), y = rnorm(50))
```

`mrm_anova_bonferroni` *One-way ANOVA with pairwise Bonferroni-adjusted t-tests*

Description

One-way ANOVA with pairwise Bonferroni-adjusted t-tests

Usage

```
mrm_anova_bonferroni(data, response_col, group_col, alpha = 0.05)
```

Arguments

`data`	data.frame.
`response_col`	Response column name.
`group_col`	Group column name.
`alpha`	Family-wise error rate (default 0.05).

Value

Named list with f_statistic, p_value, n_groups, n_pairs, alpha, alpha_per_pair, pairs (data.frame), interpretation.

Examples

```
set.seed(2026)
n <- 30L
df <- data.frame(
  y = c(rnorm(n, 0), rnorm(n, 0.5), rnorm(n, 1)),
  g = rep(c("A", "B", "C"), each = n)
)
res <- mrm_anova_bonferroni(df, response_col = "y", group_col = "g")
res$alpha_per_pair # Bonferroni-corrected per-pair alpha
res$pairs # per-pair t-tests with adjusted significance flags
```

---

mrm_anova_oneway *One-way ANOVA + Tukey HSD post-hoc*

Description

One-way ANOVA + Tukey HSD post-hoc

Usage

```
mrm_anova_oneway(data, response_col, group_col, alpha = 0.05)
```

Arguments

`data`	data.frame containing response and group columns.
`response_col, group_col`	Column names.
`alpha`	CI level (default 0.05).

Value

Named list with f_statistic, p_value, df_between, df_within, means, n_per_group, tukey_hsd, interpretation.

Examples

```
set.seed(2026)
n <- 30L
df <- data.frame(
  y = c(rnorm(n, 0), rnorm(n, 0.5), rnorm(n, 1)),
  g = rep(c("A", "B", "C"), each = n)
)
res <- mrm_anova_oneway(df, response_col = "y", group_col = "g")
res$f_statistic
```

```
res$p_value
res$tukey_hsd
```

`mrm_anova_power` *Power of one-way ANOVA given Cohen's f*

Description

Power of one-way ANOVA given Cohen's f

Usage

```
mrm_anova_power(k_groups, n_per_group, effect_size_f, alpha = 0.05)
```

Arguments

`k_groups`	Number of groups.
`n_per_group`	Per-group sample size.
`effect_size_f`	Cohen's f.
`alpha`	Type-I error (default 0.05).

Value

Named list with k_groups, n_per_group, N_total, effect_size_f, alpha, df1, df2, noncentrality, F_critical, power, interpretation.

Examples

```
# Power to detect a medium effect (Cohen's f = 0.25) with 4 groups
# of 30 each at alpha = 0.05:
res <- mrm_anova_power(
  k_groups = 4, n_per_group = 30,
  effect_size_f = 0.25, alpha = 0.05
)
res$power
res$F_critical

# Sample-size sensitivity: what power do I get with smaller groups?
sapply(c(10, 20, 30, 50, 100), function(n) {
  mrm_anova_power(
    k_groups = 3, n_per_group = n,
    effect_size_f = 0.25
  )$power
})
```

`mrm_arsau` *Per-record-type ARSAU analysis pipelines (R-side)*

Description

Six analyzers that load one ARSAU dataset via the loaders in `R/arsau.R` and chain the generic MRM Use-of-Force callables from `R/mrm_uof.R`, producing a single named list with multi-paragraph interpretation, the loaded data, all sub-analyses, and the source sidecar (if present).

Details

Functions:

- `morie_arsau_analyze_main_records`
- `morie_arsau_analyze_individual_records`
- `morie_arsau_analyze_probe_cycle_records`
- `morie_arsau_analyze_weapon_records`
- `morie_arsau_analyze_aggregate_summary`
- `morie_arsau_analyze_detailed_dataset`

Each analyzer accepts the same `year` / `language` / `data_dir` arguments as the matching loader, and returns a named list whose constituent sub-results are available under named keys (`force_concentration`, `disparity_by_race`, etc.).

`mrm_assumptions_check`
Composite Rubin-style identifiability assumption check

Description

Returns each assumption with diagnostic evidence + a flag.

Usage

```
mrm_assumptions_check(data, treatment_col, outcome_col, covariates)
```

Arguments

`data`	data.frame.
`treatment_col`	Binary 0/1 column.
`outcome_col`	Outcome column (presently unused; reserved for future E-value evidence).
`covariates`	Character vector of covariate columns.

Value

Named list with sutva, unconfoundedness, probabilistic_assignment, overall_verdict sublists.

Examples

```
set.seed(2026)
n <- 300L
x <- rnorm(n)
D <- rbinom(n, 1, plogis(0.5 * x))
y <- 0.7 * D + 0.3 * x + rnorm(n)
df <- data.frame(D = D, y = y, age = x)
chk <- mrm_assumptions_check(df,
  treatment_col = "D",
  outcome_col = "y",
  covariates = "age"
)
chk$overall_verdict
```

---

mrm_causal_design *Designed-experiment convenience wrapper around the morie causal estimator family*

Description

Designed-experiment convenience wrapper around the morie causal estimator family

Usage

```
mrm_causal_design(
  data,
  treatment_col,
  outcome_col,
  covariates = character(0),
  estimator = c("ipw", "diff_in_means")
)
```

Arguments

`data`	data.frame with treatment, outcome, optional covariates.
`treatment_col`	Binary 0/1 treatment column.
`outcome_col`	Continuous outcome column.
`covariates`	Optional character vector of covariate columns.
`estimator`	One of `"ipw"` (Hajek IPW with logistic propensity), `"diff_in_means"` (no adjustment).

Value

Named list with estimator, estimate, se, ci_lower, ci_upper, p_value, n, n_treated, interpretation.

Examples

```
set.seed(2026)
n <- 200L
x <- rnorm(n)
D <- rbinom(n, 1, plogis(0.5 * x))
y <- 0.7 * D + 0.3 * x + rnorm(n, 0, 0.5)
df <- data.frame(D = D, y = y, age = x)
# IPW-adjusted ATE
ipw <- mrm_causal_design(df,
  treatment_col = "D",
  outcome_col = "y",
  covariates = "age",
  estimator = "ipw"
)
# Naive difference in means for comparison
raw <- mrm_causal_design(df,
  treatment_col = "D",
  outcome_col = "y",
  estimator = "diff_in_means"
)
c(ipw = ipw$estimate, raw = raw$estimate)
```

`mrm_check_balancing` *Composite balance verdict using the Imbens-Rubin %SMD criterion*

Description

A design is "balanced on X" if every |SMD(X_i)| <= threshold.

Usage

```
mrm_check_balancing(data, treatment_col, covariates, threshold_pct = 10)
```

Arguments

`data`	data.frame.
`treatment_col`	Binary 0/1 treatment column.
`covariates`	Character vector of covariate columns.
`threshold_pct`	%SMD imbalance threshold (default 10).

Value

Named list with table, threshold_pct, n_imbalanced, overall_balanced, interpretation.

Examples

```
set.seed(2026)
n <- 200L
df <- data.frame(
  D   = rbinom(n, 1, 0.4),
  age = rnorm(n, 50, 10),
  bmi = rnorm(n, 27, 4)
)
df$age[df$D == 1] <- df$age[df$D == 1] + 3 # imbalance on age
bal <- mrm_check_balancing(df,
  treatment_col = "D",
  covariates = c("age", "bmi")
)
bal$overall_balanced
bal$interpretation
```

---

mrm_check_overlap *Propensity-score support overlap diagnostic (Cole-Hernan 2008)*

Description

Propensity-score support overlap diagnostic (Cole-Hernan 2008)

Usage

```
mrm_check_overlap(data, treatment_col, covariates)
```

Arguments

`data`	data.frame.
`treatment_col`	Binary 0/1 treatment column.
`covariates`	Character vector of covariates.

Value

Named list with e_treated_quantiles, e_control_quantiles, common_support_lower, common_support_upper, n_outside_support, positivity_violations, interpretation.

Examples

```
set.seed(2026)
n <- 300L
x <- rnorm(n)
D <- rbinom(n, 1, plogis(0.5 * x))
df <- data.frame(D = D, age = x)
ovl <- mrm_check_overlap(df,
  treatment_col = "D",
  covariates = "age"
)
```

```
ovl$positivity_violations
ovl$interpretation
```

`mrm_classify_mandela` *Mandela Rules classifier for solitary-confinement placements*

Description

Classify placement records under the United Nations Standard Minimum Rules for the Treatment of Prisoners (the Nelson Mandela Rules, UN A/RES/70/175) which define prolonged solitary confinement as any continuous placement exceeding fifteen days. Provides three denominator conventions and an optional broader-restrictive-confinement classification that adds high-alert placements to the numerator.

Usage

```
mrm_classify_mandela(
  data,
  duration_col = "NumberConsecutiveDays_Segregation",
  year_col = "EndFiscalYear",
  id_col = "UniqueIndividual_ID",
  threshold_days = 15L,
  denominator = c("individual_any", "row", "individual_cumulative"),
  broader_rc = FALSE,
  alert_cols = c("MentalHealth_Alert", "SuicideRisk_Alert", "SuicideWatch_Alert"),
  meaningful_contact_col = NULL
)
```

Arguments

`data`	A data.frame or data.table containing at minimum the placement-duration column and a fiscal-year column.
`duration_col`	Column name (character) of consecutive-day placement durations. Default `"NumberConsecutiveDays_Segregation"`.
`year_col`	Column name of the fiscal-year identifier. Default `"EndFiscalYear"`.
`id_col`	Column name of the per-year individual identifier. Default `"UniqueIndividual_ID"`. Required when `denominator` is anything other than `"row"`.
`threshold_days`	Mandela duration threshold in days. Default `15` (UN Standard Minimum Rules).
`denominator`	One of `"row"` (per-placement), `"individual_any"` (proportion of individuals with any placement above threshold), or `"individual_cumulative"` (proportion of individuals whose total within-year segregation days exceed threshold). Default `"individual_any"`.
`broader_rc`	Logical. If `TRUE`, the numerator additionally counts placements with alert-complexity `>= 2` (using the three alert columns named in `alert_cols`) regardless of meeting `threshold_days`. Default `FALSE`.

`alert_cols` Character vector of binary alert columns used to compute alert-complexity for the broader rate. Default the three b01 alert columns.

`meaningful_contact_col` Optional. Column name of a 1-if-meaningful-contact indicator (federal Sprott-Doob style). When supplied, rows with met-contact are excluded from the numerator.

Details

The provincial classification operates on duration alone (the `duration_col` column). The federal classification additionally requires unmet "meaningful contact" criteria (Sprott & Doob, 2021); if `meaningful_contact_col` is supplied, that column is treated as a 1-if-met indicator and rows with met-contact are excluded from the numerator regardless of duration.

Value

A data.frame with columns:

year Fiscal year (or `"pooled"`).

denominator Total denominator under the chosen convention.

n_mandela Numerator: count of records (or individuals) classified as Mandela-prolonged.

rate Proportion `n_mandela / denominator`, in the unit interval.

pct Same as `rate` expressed as percentage.

n_broader_rc Broader-rate numerator (if `broader_rc`).

rate_broader Broader-rate proportion (if `broader_rc`).

References

United Nations General Assembly (2015). United Nations Standard Minimum Rules for the Treatment of Prisoners (the Nelson Mandela Rules). A/RES/70/175.

Sprott, J. B., & Doob, A. N. (2021). Solitary Confinement, Torture, and Canada's Structured Intervention Units. Centre for Criminology and Sociolegal Studies, University of Toronto. Available at the Centre for Criminology and Sociolegal Studies web site: crimsl.utoronto.ca (file TortureSolitarySIUsSprottDoob23Feb2021_0.pdf).

Iftene, A., & Doob, A. N. (2024). Do Independent External Decision Makers Ensure that "An Inmate's Confinement in a Structured Intervention Unit Is to End as Soon as Possible"? (Corrections and Conditional Release Act, Section 33). Dalhousie Schulich School of Law, report 51. https://digitalcommons.schulichlaw.dal.ca/reports/51/

Examples

```
# Strict provincial Mandela on b01:
#   mrm_classify_mandela(b01_data)
#
# Broader restrictive-confinement (adds high-alert placements):
#   mrm_classify_mandela(b01_data, broader_rc = TRUE)
```

`mrm_clt_demo` *Central Limit Theorem demonstrator*

Description

Generate sample means from a base distribution.

Usage

```
mrm_clt_demo(
  base_distribution = "unif",
  n_samples = 1000L,
  sample_size = 30L,
  seed = 42L,
  ...
)
```

Arguments

`base_distribution`	Distribution suffix passed to `r<dist>` (e.g. `"unif"`, `"exp"`, `"pois"`).
`n_samples`	Number of sample means.
`sample_size`	Size of each sample.
`seed`	RNG seed.
`...`	Additional parameters passed to `r<dist>`.

Value

data.frame with sample_index, sample_mean, z_score.

Examples

```
# 1000 sample means of size 30 from an exponential(1) base;
# standardised z-scores converge to N(0,1):
res <- mrm_clt_demo(
  base_distribution = "exp",
  n_samples = 1000L,
  sample_size = 30L,
  seed = 42L, rate = 1
)
summary(res$z_score)
# mean ~ 0, sd ~ 1
```

mrm_design *Experimental-design callables (designexptr-inspired)*

Description

R parity of `morie.mrm_design` (Python). Four general- purpose statistical-design entry points covering the designexptr.org pedagogical sequence: two-treatment comparison, one-way ANOVA with Tukey HSD, 2^k factorial design, and a designed-experiment convenience wrapper around the morie causal estimator family.

Value

Each design callable returns a named `list` of estimates, test statistics, p-values, and a plain-language `interpretation`.

References

Box, G. E. P., Hunter, J. S., & Hunter, W. G. (2005). Statistics for Experimenters. Wiley.

Examples

```
set.seed(2026)
a <- rnorm(40, mean = 5, sd = 1.2)
b <- rnorm(40, mean = 5.5, sd = 1.5)
mrm_two_treatment_test(a, b)$p_welch
```

mrm_diagnostics *Causal-inference diagnostics (R parity)*

Description

Balance, overlap, SUTVA-style assumption checks, and the median causal effect estimator. R parity of `morie.mrm_diagnostics`.

Value

Each diagnostic callable returns a named `list` of balance and overlap statistics (or the estimated effect) together with a plain-language `interpretation`.

References

Imbens, G. W., & Rubin, D. B. (2015). Causal Inference for Statistics, Social and Biomedical Sciences. Cambridge University Press. Rosenbaum, P. R., & Rubin, D. B. (1985). Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. The American Statistician, 39(1), 33-38. Cole, S. R., & Hernan, M. A. (2008). Constructing inverse probability weights for marginal structural models. AJE, 168(6), 656-664.

Examples

```
set.seed(2026)
n <- 200L
df <- data.frame(
  D = rbinom(n, 1, 0.4),
  age = rnorm(n, 50, 10), bmi = rnorm(n, 27, 4)
)
mrm_standardised_difference(df,
  treatment_col = "D",
  covariates = c("age", "bmi")
)
```

`mrm_doe` *Design-of-Experiments toolkit (R parity)*

Description

R parity of `morie.mrm_doe`. Closes the Chapter-3/4/5 coverage gap from designexptr.org.

Value

Each design-of-experiments callable returns a named `list` holding the constructed design or the analysis result and a plain-language `interpretation`.

References

Box, G. E. P., Hunter, J. S., & Hunter, W. G. (2005). Statistics for Experimenters (2nd ed.). Wiley. Cochran, W. G., & Cox, G. M. (1957). Experimental Designs (2nd ed.). Wiley. Montgomery, D. C. (2017). Design and Analysis of Experiments (9th ed.). Box, G. E. P., & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. JRSS-B, 13(1), 1-45. Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences.

Examples

```
set.seed(2026)
n <- 30L
df <- data.frame(
  y = c(rnorm(n, 0), rnorm(n, 0.5), rnorm(n, 1)),
  g = rep(c("A", "B", "C"), each = n)
)
mrm_anova_bonferroni(df, response_col = "y", group_col = "g")$alpha_per_pair
```

`mrm_factorial_2k` *2^k factorial-design analysis with main effects + interactions*

Description

Returns main effects (difference of means at +1 vs -1 per factor), all interaction effects, and half-normal-plot coordinates for Daniel's method (which lets the user separate active effects from a null half-normal line on the same axes).

Usage

```
mrm_factorial_2k(data, response_col, factor_cols)
```

Arguments

`data`	data.frame with response_col and factor_cols. Factor columns may be coded as `+/- 1` or any binary; non-(-1,1) columns are re-coded.
`response_col`	Numeric response column.
`factor_cols`	Character vector of factor column names.

Value

Named list with main_effects, interaction_effects, half_normal_coords (data.frame), n, k, interpretation.

Examples

```
# 2^3 full factorial: 8 runs, factors A, B, C in {-1, +1}.
set.seed(2026)
lvl <- c(-1, 1)
df <- expand.grid(A = lvl, B = lvl, C = lvl)
df$y <- 10 + 2 * df$A + 1.5 * df$B + 0.5 * df$A * df$B + rnorm(8, 0, 0.2)
res <- mrm_factorial_2k(df,
  response_col = "y",
  factor_cols = c("A", "B", "C")
)
res$main_effects
res$interaction_effects
```

`mrm_fractional_factorial` *Fractional 2^(k-p) factorial: main effects + alias structure*

Description

Factor columns assumed +/-1.

Usage

```
mrm_fractional_factorial(data, response_col, factor_cols, generator = NULL)
```

Arguments

`data`	data.frame.
`response_col`	Response column.
`factor_cols`	Character vector of factor columns (each coded -1 or +1).
`generator`	Optional generator string "X=YZ,..." for aliasing.

Value

Named list with main_effects, alias_structure, n, k, interpretation.

Examples

```
# 2^(3-1) fractional with D = A*B*C generator: 4 runs instead of 8.
set.seed(2026)
lvl <- c(-1, 1)
df <- data.frame(
  A = c(-1, 1, -1, 1),
  B = c(-1, -1, 1, 1),
  C = c(1, -1, -1, 1)
)
df$y <- 5 + 2 * df$A + 1.5 * df$B + rnorm(4, 0, 0.3)
res <- mrm_fractional_factorial(df,
  response_col = "y",
  factor_cols = c("A", "B", "C")
)
res$main_effects
```

mrm_gentrification *Baseline-conditional gentrification coding (MRM primitive)*

Description

Three-level categorical coding of tract-level gentrification, mirroring the Python module `morie.mrm_primitives.gentrification`. Adapted from Laniyonu (2018) Urban Affairs Review 54(5):898-930, which itself adapts Chapple / Freeman / Maciag.

Details

The key insight: continuous gentrification indices conflate two distinct populations – already-affluent tracts (immune to gentrification by construction) and marginalised tracts that DID or DID NOT change. The cleanest comparator is the marginalised-but- did-not-gentrify tract, so this primitive emits a 3-level factor:

- `ineligible` – tract was above the baseline- marginalisation cutoff (top-50\ cannot meaningfully "gentrify". Drop from analyses that want the gentrification comparator.
- `eligible` – tract was below the cutoff at t=0 AND did NOT cross the gentrification threshold by t=1. This is the control: marginalised, did-not-change.
- `gentrified` – tract was below the cutoff at t=0 AND DID cross the gentrification threshold (top-tercile growth in college share AND top-tercile growth in median rent).

mrm_gentrification_panel *Construct a baseline-conditional 3-level gentrification factor*

Description

Implements the Laniyonu (2018) operationalisation:

Usage

```
mrm_gentrification_panel(
  df,
  baseline_income_col,
  baseline_rent_col,
  growth_college_col,
  growth_rent_col,
  baseline_marginalisation_quantile = 0.5,
  gentrification_growth_quantile = 0.667
)
```

Arguments

`df` A `data.frame` with one row per tract; must contain the four named columns.

`baseline_income_col` Character. Column carrying baseline (period t=0) income.

`baseline_rent_col` Character. Column carrying baseline rent.

`growth_college_col` Character. Column carrying college / BA-share growth between baseline and follow-up.

`growth_rent_col` Character. Column carrying median-rent growth between baseline and follow-up.

`baseline_marginalisation_quantile` Numeric in (0, 1); default 0.5. Tract is eligible if baseline income AND rent are $\leq$ this quantile.

`gentrification_growth_quantile` Numeric in (0, 1); default 0.667. Tract gentrifies if college growth AND rent growth are $\geq$ this quantile.

Details

1. Tract is *eligible* to gentrify iff baseline income AND baseline rent are at or below `baseline_marginalisation_quantile` of the panel.
2. Among the eligible, the tract is *gentrified* iff growth-in-college-share AND growth-in-rent are at or above `gentrification_growth_quantile`.
3. Everything above the baseline cut is *ineligible*.

Value

A named list with classes `morie_mrm_result`, `morie_rich_result`, `list`. Carries `labels` (character vector of length `nrow(df)`), `thresholds` (list of four cut-points), `counts` (table of label levels), plus `interpretation` + `warnings`.

Examples

```
set.seed(1)
df <- data.frame(
  inc0  = runif(50, 20000, 80000),
  rent0 = runif(50, 500, 2000),
  coll_g = rnorm(50),
  rent_g = rnorm(50)
)
res <- mrm_gentrification_panel(
  df,
  baseline_income_col = "inc0",
  baseline_rent_col   = "rent0",
  growth_college_col  = "coll_g",
  growth_rent_col     = "rent_g"
```

```
)
table(res$labels)
```

`mrm_graeco_latin` *Graeco-Latin square four-way ANOVA (row, col, Latin, Greek)*

Description

Graeco-Latin square four-way ANOVA (row, col, Latin, Greek)

Usage

```
mrm_graeco_latin(data, response_col, row_col, col_col, latin_col, greek_col)
```

Arguments

`data`	data.frame.
`response_col`, `row_col`, `col_col`, `latin_col`, `greek_col`	Column names.

Value

Named list with anova, n, interpretation.

Examples

```
# Hardcoded 4 x 4 orthogonal Graeco-Latin square (two random Latin
# squares are generally NOT orthogonal, so we use a known pair):
L <- matrix(c(
  "A", "B", "C", "D",
  "B", "A", "D", "C",
  "C", "D", "A", "B",
  "D", "C", "B", "A"
), nrow = 4L, byrow = TRUE)
G <- matrix(c(
  "a", "b", "c", "d",
  "c", "d", "a", "b",
  "d", "c", "b", "a",
  "b", "a", "d", "c"
), nrow = 4L, byrow = TRUE)
set.seed(2026)
df <- expand.grid(row = paste0("R", 1:4), col = paste0("C", 1:4))
df$latin <- as.vector(L)
df$greek <- as.vector(G)
df$y <- match(df$latin, LETTERS) * 1.2 +
  match(df$greek, letters) * 0.5 + rnorm(16, 0, 0.3)
res <- mrm_graeco_latin(df,
  response_col = "y",
  row_col = "row", col_col = "col",
  latin_col = "latin", greek_col = "greek"
```

```
)
res$anova
```

`mrm_kulldorff` *Kulldorff space-time scan statistic on TPS event data*

Description

R parity of `morie.mrm_kulldorff.mrm_tps_kulldorff_scan()`. Implements Kulldorff's 1997 Poisson log-likelihood-ratio space-time scan with a Monte-Carlo permutation test for significance.

Details

The scan iterates over (centre, radius, time-window) tuples, computing the Poisson LRT against H_0 (events uniformly distributed in space and time). The maximum LRT is the test statistic; permutations of event timestamps generate the null.

Value

`mrm_tps_kulldorff_scan()` returns a named `list` with the most likely cluster, its Poisson log-likelihood-ratio statistic, the Monte-Carlo permutation p-value, and a plain-language `interpretation`.

References

Kulldorff, M. (1997). A spatial scan statistic. Communications in Statistics: Theory and Methods, 26(6), 1481–1496.

Examples

```
if (FALSE) {
  tps <- morie_sample("tps_assault")
  mrm_tps_kulldorff_scan(tps, n_permutations = 49)
}
```

`mrm_latin_square` *Latin-square three-way ANOVA (row, col, treatment)*

Description

Latin-square three-way ANOVA (row, col, treatment)

Usage

```
mrm_latin_square(data, response_col, row_col, col_col, treatment_col)
```

Arguments

`data`	data.frame.
`response_col`, `row_col`, `col_col`, `treatment_col`	Column names.

Value

Named list with anova, n, k, interpretation.

Examples

```
# 4 x 4 Latin square: each treatment appears once per row and column.
# `mrm_random_latin()` returns integer codes 0..k-1; convert to
# letters for a more readable example.
sq <- mrm_random_latin(k = 4, seed = 2026)
df <- expand.grid(row = paste0("R", 1:4), col = paste0("C", 1:4))
df$treatment <- LETTERS[as.integer(as.vector(sq)) + 1L]
set.seed(2026)
df$y <- match(df$treatment, LETTERS) * 1.5 + rnorm(16, 0, 0.4)
res <- mrm_latin_square(df,
  response_col = "y",
  row_col = "row", col_col = "col",
  treatment_col = "treatment"
)
res$anova
```

`mrm_lisa` *LISA (Local Indicators of Spatial Association) on polygon-level crime data + per-year polygon Moran's I time series*

Description

R parity of `morie.mrm_tps_lisa()` and `morie.mrm_tps_polygon_moran_per_year()`. Local Moran's I per polygon centroid with 999-permutation MC significance, plus a convenience wrapper for the per-year time series used by the morie empirical paper Section 7.11.

Value

The LISA callables return named `list`s with per-polygon local Moran's I, permutation p-values, cluster classifications, and (for the per-year wrapper) the time series of global Moran's I.

References

Anselin, L. (1995). Local indicators of spatial association – LISA. *Geographical Analysis*, 27(2), 93–115.

Examples

```
if (FALSE) {
  ncr <- read.csv("Neighbourhood_Crime_Rates_Open_Data.csv")
  mrm_tps_lisa(ncr,
    count_col = "ASSAULT_2024",
    lat_col = "lat", lon_col = "lon"
  )
}
```

---

mrm_mathstats — *Mathematical-statistics / simulation / computation toolkit (R parity)*

Description

R parity of `morie.mrm_mathstats`. Closes the Chapter-2 coverage gap from designexptr.org/mathematical-statistics- simulation-and-computation.html.

Value

Each callable returns a named `list` with the computed statistic(s) and a plain-language `interpretation`.

References

Wilks, S. S. (1962). Mathematical Statistics. Wiley. Casella, G. & Berger, R. L. (2002). Statistical Inference. Duxbury. Lehmann, E. L. & Romano, J. P. (2005). Testing Statistical Hypotheses.

Examples

```
mrm_oneprop_test(x = 58, n = 100, p0 = 0.5)
```

---

mrm_mc_power — *Empirical Monte-Carlo power*

Description

Empirical Monte-Carlo power

Usage

```
mrm_mc_power(simulator, n_sims = 1000L, alpha = 0.05, seed = 42L)
```

Arguments

simulator	A function(seed) returning a p-value.
n_sims	Number of simulated datasets.
alpha	Type-I error level.
seed	Seed for outer RNG.

Value

Named list with empirical_power, se, ci95 bounds.

Examples

```
# Empirical power of a one-sample t-test against H0: mu = 0
# with true mu = 0.4 and n = 30.
my_sim <- function(seed) {
  set.seed(seed)
  x <- rnorm(30, mean = 0.4, sd = 1)
  stats::t.test(x, mu = 0)$p.value
}
res <- mrm_mc_power(my_sim, n_sims = 500L, alpha = 0.05)
res$empirical_power
res$ci95_lower
res$ci95_upper
```

---

mrm_median_causal_effect

Median causal effect via 1:1 nearest-neighbour PS matching

Description

Median causal effect via 1:1 nearest-neighbour PS matching

Usage

```
mrm_median_causal_effect(data, treatment_col, outcome_col, covariates)
```

Arguments

data	data.frame.
treatment_col	Binary 0/1 column.
outcome_col	Outcome column name.
covariates	Character vector of covariate columns.

Value

Named list with median_y1, median_y0, median_treatment_effect, n_matched, interpretation.

Examples

```
set.seed(2026)
n <- 200L
x <- rnorm(n)
D <- rbinom(n, 1, plogis(0.5 * x))
y <- 0.7 * D + 0.3 * x + rnorm(n, 0, 0.5)
df <- data.frame(D = D, y = y, age = x)
res <- mrm_median_causal_effect(df,
  treatment_col = "D",
  outcome_col = "y",
  covariates = "age"
)
res$median_treatment_effect
res$n_matched
```

---

mrm_morans_i *Moran's I statistic for residual spatial autocorrelation*

Description

First rung of the diagnostic ladder: if OLS residuals show significant Moran's I, an SDM (or SEM/SAR) is warranted over OLS.

Usage

```
mrm_morans_i(residuals, W)
```

Arguments

`residuals`	Numeric vector of length N (e.g. OLS residuals).
`W`	Numeric matrix of shape (N, N) – the spatial weight matrix. Need not be row-standardised but must be aligned with `residuals`.

Details

Statistic:

$$I = \frac{n}{\sum_{ij} w_{ij}} \cdot \frac{e^\top W e}{e^\top e}, \quad e = r - \bar{r}.$$

$I \in$ '$[-1, 1]$'. Positive -> clustering, negative -> dispersion, ~0 -> spatial randomness.

Value

A named list with classes `morie_mrm_result`, `morie_rich_result`, `list`. Carries `morans_i` (the scalar statistic) plus `interpretation`

- `warnings`.

Examples

```
set.seed(4)
N <- 20
W <- matrix(runif(N * N), N, N); diag(W) <- 0; W <- W / rowSums(W)
resid <- rnorm(N)
mrm_morans_i(resid, W)$morans_i
```

---

mrm_oneprop_test *One-proportion test (binomial exact + Wald approximation)*

Description

One-proportion test (binomial exact + Wald approximation)

Usage

```
mrm_oneprop_test(x, n, p0, alpha = 0.05)
```

Arguments

x	Number of successes.
n	Number of trials.
p0	Null-hypothesis proportion.
alpha	CI level (default 0.05 -> 95% CI).

Value

Named list with p_hat, p0, n, z_wald, p_value_wald, p_value_exact, ci95_wald_lower/upper, ci95_exact_lower/upper, interpretation.

Examples

```
# H0: proportion = 0.5 against the observed 58/100 successes
mrm_oneprop_test(x = 58, n = 100, p0 = 0.5)
```

`mrm_otis` *MRM-framework analyses on Ontario OTIS data*

Description

Five callables for the OTIS (Offender Tracking Information System) public-release datasets, used in the MRM (Multilevel Reconciliation Methodology) empirical companion paper. Every analysis is computed directly from the OTIS CSV files; no precomputed artifacts are required.

Details

Functions:

- `mrm_otis_placement_concentration()`: Hill-MLE Pareto exponent + Gini coefficient + top-k% concentration on the b09 per-individual placement-count distribution (within-fiscal-year).
- `mrm_otis_seg_duration_km()`: Kaplan-Meier survival on the b01 `NumberConsecutiveDays_Segregation` durations (per-placement; strata = alert profile).
- `mrm_otis_mortification_cooccurrence()`: pairwise Cramer's V across the three b01 alert flags (MentalHealth, SuicideRisk, SuicideWatch).
- `mrm_otis_region_locality()`: chi-square + Cramer's V on the `Region_AtTimeOfPlacement` x `Region_MostRecentPlacement` contingency table, with the diagonal/off-diagonal share.
- Plus the existing `mrm_classify_mandela()` (in `mandela.R`).

The OTIS `UniqueIndividual_ID` column has format `YYYY-XXXXX-SG` and is randomly reassigned every fiscal year. Cross-year tracking is therefore invalid; all analyses below operate within fiscal year.

Value

Each `mrm_otis_*()` callable returns a named `list` with the computed statistics (concentration indices, survival curves, or association measures) and a plain-language `interpretation`.

Examples

```
if (FALSE) {
  b09 <- read.csv("b09_individuals_in_segregation.csv")
  mrm_otis_placement_concentration(b09)
}
```

`mrm_otis_mandela_spectrum` *Mandela Rules apples-to-apples spectrum on OTIS b01*

Description

R parity of `morie.mrm_otis_mandela_spectrum()`. Computes a full grid of provincial Mandela-classified rates across four denominator conventions x three meaningful-contact proxies, so the *Cross-jurisdiction comparison* table in the MRM formulations paper (Section 5.3) can be reproduced from a single function call.

Usage

```
mrm_otis_mandela_spectrum(
  data,
  duration_col = "NumberConsecutiveDays_Segregation",
  year_col = "EndFiscalYear",
  id_col = "UniqueIndividual_ID",
  threshold_days = 15L,
  alert_cols = c("MentalHealth_Alert", "SuicideRisk_Alert", "SuicideWatch_Alert"),
  contact_proxies = c("none", "any_alert", "no_alert"),
  denominators = c("row", "individual_any", "individual_cumulative"),
  c11_data = NULL
)
```

Arguments

`data`	OTIS b01 data.frame.
`duration_col`, `year_col`, `id_col`	Column names.
`threshold_days`	Rule 43 duration threshold (UN: 15 days).
`alert_cols`	Three b01 alert columns (`"Yes"` = active).
`contact_proxies`	Subset of `c("none","any_alert","no_alert")`.
`denominators`	Subset of `c("row","individual_any","individual_cumulative","c11_aggregate")`.
`c11_data`	Optional c11 aggregate frame for the `c11_aggregate` denominator.

Details

Denominator conventions:

`row` per-placement rate (b01 row count)

`individual_any` share of within-year individuals with any placement satisfying the criterion

`individual_cumulative` share of within-year individuals whose cumulative within-year segregation days exceeds the threshold

`c11_aggregate` the duration-band aggregate from c11 (requires `c11_data` to be supplied)

Meaningful-contact proxies (Rule 44, derived from b01 alert flags):

`none` Rule 43 only; no contact proxy applied

`any_alert` Rule 43 AND (any of MH/SR/SW alert active); these placements receive staff contact, so this is the looser contact-failure proxy

`no_alert` Rule 43 AND (no alert active); strictest contact-failure proxy

Value

Tidy long-format data.frame with one row per (year, denominator, contact_proxy) cell, columns `year`, `denominator`, `contact_proxy`, `n_eligible`, `n_mandela`, `rate`, `pct`.

References

United Nations General Assembly (2015). United Nations Standard Minimum Rules for the Treatment of Prisoners (the Nelson Mandela Rules). A/RES/70/175. Rule 43 = prolonged (more than 15 days). Rule 44 = at least 22 hours/day, no meaningful human contact.

Examples

```
if (FALSE) {
  b01 <- read.csv("b01_segregation_detailed_dataset.csv")
  spec <- mrm_otis_mandela_spectrum(b01)
  head(spec)
}
```

`mrm_otis_mortification_cooccurrence` *Pairwise Cramer's V of OTIS b01 alert columns (mortification proxy)*

Description

Computes the pairwise Cramer's V (and chi-square test) for every pair of the three OTIS b01 alert columns. The MentalHealth x SuicideRisk Cramer's V is the substantive "mortification co-occurrence" figure used in the MRM paper.

Usage

```
mrm_otis_mortification_cooccurrence(
  data,
  alert_cols = c("MentalHealth_Alert", "SuicideRisk_Alert", "SuicideWatch_Alert")
)
```

Arguments

`data`	A data.frame with at least the three alert columns.
`alert_cols`	Character vector of alert column names (default the three b01 alert columns).

Details

Values are computed by treating `"Yes"` as 1 and any other value as 0; rows with `NA` in either alert column are dropped from that pair.

Value

A data.frame with one row per pair, columns `alert_a`, `alert_b`, `n`, `chi2`, `df`, `p_value`, `morie_cramers_v`.

Examples

```
if (FALSE) {
  b01 <- read.csv("b01_segregation_detailed_dataset.csv")
  mrm_otis_mortification_cooccurrence(b01)
}
```

---

mrm_otis_placement_concentration

Per-individual segregation-placement-count concentration on OTIS b09

Description

Expands the OTIS b09 banded per-individual placement counts into a per-person vector using band midpoints (the published bands are `\{1, 2, 3, 4, 5, 6-10, 11-15, 16-20, 21-25, 26-30, 31-35,` then computes Hill-MLE Pareto exponent, Gini coefficient, and top-k% concentration within each fiscal year and pooled.

Usage

```
mrm_otis_placement_concentration(
  data,
  year_col = "EndFiscalYear",
  band_col = "NumberPlacements_Segregation",
  count_col = "NumberIndividuals_Segregation",
  gender_col = NULL,
  gender_keep = NULL,
  x_min = 1L,
  top_pct = 0.05
)
```

Arguments

`data`	A data.frame in b09 long format with the columns named in `year_col`, `count_col`, `band_col`, optionally `gender_col`.
`year_col`	Column name of the fiscal-year identifier (default `"EndFiscalYear"`).
`band_col`	Column name of the placement-count band (default `"NumberPlacements_Segregation"`).
`count_col`	Column name of the per-band individual count (default `"NumberIndividuals_Segregation"`).
`gender_col`	Optional gender filter column. If supplied with `gender_keep`, rows are restricted to the kept genders.
`gender_keep`	Character vector of gender values to retain.
`x_min`	Hill-MLE lower-tail cutoff (default `1L`).
`top_pct`	Numeric in (0, 1); top concentration cutoff (default `0.05`).

Value

A data.frame with one row per fiscal year plus a final `"pooled"` row, containing columns `year`, `n_individuals`, `n_placements`, `mean_per_individual`, `gini`, `hill_alpha`, `top_pct_share`.

References

Hill, B. M. (1975). A simple general approach to inference about the tail of a distribution. The Annals of Statistics, 3(5), 1163-1174.

Clauset, A., Shalizi, C. R., & Newman, M. E. J. (2009). Power-law distributions in empirical data. SIAM Review, 51(4), 661-703.

Examples

```
if (FALSE) {
  b09 <- read.csv("b09_individuals_in_segregation_number_of_times_in_segregation.csv")
  mrm_otis_placement_concentration(b09)
}
```

`mrm_otis_region_locality`

OTIS b01 region locality: chi-square + diagonal-share

Description

Constructs the contingency table `Region_AtTimeOfPlacement` x `Region_MostRecentPlacement` and reports the chi-square statistic, Cramer's V, and the share of placements on the diagonal (within-region staying) vs off-diagonal (cross-region churn). Ontario seg/RC placement is overwhelmingly diagonal (locality-preserving) in the public release.

Usage

```
mrm_otis_region_locality(
  data,
  region_at_col = "Region_AtTimeOfPlacement",
  region_recent_col = "Region_MostRecentPlacement"
)
```

Arguments

`data`	A data.frame with `region_at_col` and `region_recent_col`.
`region_at_col`	Column name of the at-placement region (default `"Region_AtTimeOfPlacement"`).
`region_recent_col`	Column name of the most-recent region (default `"Region_MostRecentPlacement"`).

Value

A list with named elements `table` (the contingency matrix), `chi2`, `df`, `p_value`, `morie_cramers_v`, `diagonal_share`, `off_diagonal_share`.

Examples

```
if (FALSE) {
  b01 <- read.csv("b01_segregation_detailed_dataset.csv")
  mrm_otis_region_locality(b01)
}
```

---

mrm_otis_seg_duration_km

Kaplan-Meier survival on OTIS b01 segregation-placement durations

---

Description

Treats each row of OTIS b01 as one observed placement with duration `NumberConsecutiveDays_Segregation` and no censoring (all durations are observed end-to-end within the fiscal year). Returns the median duration and the requested-quantile survival probabilities by stratum.

Usage

```
mrm_otis_seg_duration_km(
  data,
  duration_col = "NumberConsecutiveDays_Segregation",
  group_cols = NULL,
  probs = c(0.5, 0.25, 0.1, 0.05, 0.01),
  mandela_threshold = 15L
)
```

Arguments

data	A data.frame containing duration_col and optional stratifying columns (group_cols).
duration_col	Column name of segregation duration in days (default "NumberConsecutiveDays_Segregation")
group_cols	Optional character vector of stratifying-column names. NULL pools all rows.
probs	Quantiles of the survival function to report (default c(0.5, 0.25, 0.10, 0.05, 0.01)).
mandela_threshold	Day cutoff (default 15L) for Mandela- prolonged placement. Reports the fraction of placements above the cutoff and the median duration among those.

Details

This replaces the misreading of UniqueIndividual_ID = YYYY-XXXXX-SG as a persistent person identifier (which produces a spurious ~210-day cross-year TTR artifact). The valid quantity here is the distribution of how long a placement lasts, not how long until the next placement.

Value

A data.frame with one row per stratum (or one pooled row), columns stratum, n, mean_days, median_days, q25_days, pct_above_mandela, median_among_above_mandela.

Examples

```
if (FALSE) {
  b01 <- read.csv("b01_segregation_detailed_dataset.csv")
  mrm_otis_seg_duration_km(b01)
  mrm_otis_seg_duration_km(b01, group_cols = "MentalHealth_Alert")
}
```

---

mrm_perm_block *Block-permutation test for treatment effect*

Description

Permutes treatment labels within each block.

Usage

```
mrm_perm_block(
  data,
  response_col,
  treatment_col,
  block_col,
```

```
  n_perm = 1000L,
  seed = 42L
)
```

Arguments

`data`	data.frame.
`response_col`, `treatment_col`, `block_col`	Column names.
`n_perm`	Number of permutations.
`seed`	RNG seed.

Value

Named list with observed_statistic, n_perm, p_value, interpretation.

Examples

```
set.seed(2026)
df <- expand.grid(
  block = paste0("B", 1:6),
  treatment = c("ctrl", "drug")
)
# Block-level baseline + treatment effect
df$y <- as.numeric(df$block) * 1.2 +
  ifelse(df$treatment == "drug", 0.7, 0) +
  rnorm(nrow(df), 0, 0.4)
res <- mrm_perm_block(df,
  response_col = "y",
  treatment_col = "treatment",
  block_col = "block",
  n_perm = 500L
)
res$p_value
```

`mrm_pit` *Probability Integral Transform (PIT)*

Description

If X ~ F, then F(X) ~ Uniform(0,1). Returned U should be approx uniform if the assumed F is correct. Attaches a KS p-value of U against Uniform(0,1) as the diagnostic for fit quality.

Usage

```
mrm_pit(sample, dist = "norm", ...)
```

Arguments

`sample`	Numeric vector.
`dist`	Distribution suffix for `p<dist>`.
`...`	Additional parameters for `p<dist>`.

Value

data.frame with raw, U columns and attributes ks_stat, ks_pvalue.

Examples

```
set.seed(2026)
x <- rnorm(200)
# Under correct distributional assumption, U should be ~Uniform(0,1):
pit <- mrm_pit(x, dist = "norm")
attr(pit, "ks_pvalue") # large p-value => no evidence against fit
# If we deliberately misspecify (claim t_3 fits the normal sample):
pit_wrong <- mrm_pit(x, dist = "t", df = 3)
attr(pit_wrong, "ks_pvalue") # small p-value => misspecification detected
```

`mrm_primitives_ordinal`

Threshold-specific ordinal-logit primitive (MRM)

Description

R parity of `morie.mrm_primitives.threshold_specific_ordinal()`. Adapted from O'Connell & Laniyonu (2025) *Race & Justice* 15(3):428–453, where a Bayesian cumulative-logit model is fit with race / gender coefficients allowed to VARY by cumulative threshold. The empirically critical finding – bias concentrated at the low->medium cutoff but not the medium->high cutoff – is invisible to standard proportional-odds specifications.

Details

This R port is the **frequentist** analogue: for each cutpoint $k = 1, \ldots, K-1$ a separate binary logit is fit to the indicator $1\{Y \le k\}$, so the coefficient vector β_k is unconstrained across thresholds. When `MASS` is available we delegate to `polr` for the proportional-odds (PO) baseline; otherwise the PO baseline is fit by a stacked-IRLS approximation matching the Python implementation. The threshold-specific fits always run via `glm` with `family = binomial("logit")`.

Standard threshold (proportional-odds, K levels, p covariates):

$$P(Y \le k \mid X) = \text{logit}^{-1}(\alpha_k - X\beta)$$

Threshold-specific extension (one coefficient vector per cutpoint):

$$P(Y \le k \mid X) = \text{logit}^{-1}(\alpha_k - X\beta_k)$$

References

O'Connell, M. & Laniyonu, A. (2025). Threshold-specific cumulative-logit models for actuarial-risk audit. *Race & Justice*, 15(3), 428–453.

See Also

`mrm_score_net_residual` (internal helper)

`mrm_qq_plot` *Q-Q plot coordinates against a reference distribution*

Description

Q-Q plot coordinates against a reference distribution

Usage

```
mrm_qq_plot(sample, dist = "norm", ...)
```

Arguments

`sample`	Numeric vector.
`dist`	Either `"norm"`, `"exp"`, `"unif"`, `"t"`, `"chisq"` (any base `q<dist>` works).
`...`	Additional parameters passed to `q<dist>`.

Value

data.frame with rank, empirical, theoretical, plotting_position columns (Blom 1958 plotting positions).

Examples

```
set.seed(2026)
x <- rnorm(100)
qq <- mrm_qq_plot(x, dist = "norm")
head(qq)
# plot(qq$theoretical, qq$empirical); abline(0, 1)
```

mrm_random_latin *Generate a random k x k Latin square*

Description

Builds the cyclic Latin square then permutes rows, columns, and symbols. Uniform over a subset of Latin squares (not all).

Usage

```
mrm_random_latin(k, seed = 42L)
```

Arguments

k	Side length.
seed	RNG seed.

Value

A k x k integer matrix (codes 0..k-1) with row names R1..Rk and column names C1..Ck. Each code appears exactly once per row and per column.

Examples

```
# 4 x 4 random Latin square: each of {0, 1, 2, 3} appears once
# per row and per column.
mrm_random_latin(k = 4, seed = 42L)

# Reproducible across runs with the same seed:
identical(
  mrm_random_latin(5, seed = 7),
  mrm_random_latin(5, seed = 7)
)
```

mrm_rcbd *Randomised complete block design (RCBD) two-way ANOVA*

Description

Model: y_ij = mu + tau_i (treatment) + beta_j (block) + eps_ij Returns Type-I ANOVA: block enters first, then treatment.

Usage

```
mrm_rcbd(data, response_col, treatment_col, block_col)
```

Arguments

`data`	data.frame.
`response_col`, `treatment_col`, `block_col`	Column names.

Value

Named list with anova (data.frame), n, n_treatments, n_blocks, interpretation.

Examples

```
set.seed(2026)
df <- expand.grid(
  treatment = c("A", "B", "C"),
  block = c("B1", "B2", "B3", "B4")
)
# Treatment effect + block effect + noise
df$y <- as.numeric(df$treatment) * 2 +
  as.numeric(df$block) * 0.5 + rnorm(nrow(df), 0, 0.3)
res <- mrm_rcbd(df,
  response_col = "y",
  treatment_col = "treatment", block_col = "block"
)
res$anova
```

`mrm_response_surface` *Second-order response-surface fit (Box-Wilson 1951)*

Description

Fits y = b0 + sum b_i x_i + sum b_ii x_i^2 + sum b_ij x_i x_j and returns the stationary point if the quadratic matrix B is invertible.

Usage

```
mrm_response_surface(data, response_col, factor_cols)
```

Arguments

`data`	data.frame.
`response_col`	Response column.
`factor_cols`	Character vector of factor columns.

Value

Named list with coefficients, stationary_point, stationary_y, stationary_nature, eigenvalues, n, interpretation.

Examples

```
# Central composite design on (x1, x2) with quadratic response.
set.seed(2026)
df <- expand.grid(
  x1 = c(-1.4, -1, 0, 1, 1.4),
  x2 = c(-1.4, -1, 0, 1, 1.4)
)
df$y <- 10 + 2 * df$x1 + 1.5 * df$x2 -
  df$x1^2 - 1.2 * df$x2^2 + rnorm(nrow(df), 0, 0.2)
res <- mrm_response_surface(df,
  response_col = "y",
  factor_cols = c("x1", "x2")
)
res$stationary_point
res$stationary_nature
```

---

mrm_samples *Bundled reference data samples and dataset fetchers*

Description

MORIE ships a small set of reference CSVs in `inst/extdata/` so that the `mrm_otis_*()` and `mrm_tps_*()` callables can be exercised without any network call. For full datasets, the on-demand fetchers pull from the original public sources:

Details

- OTIS: data.ontario.ca CKAN package `data-on-inmates-in-ontario`. Resource IDs are baked into `morie_dataset_catalog()`; use `morie_load_dataset("otisb01")` (etc.) which calls the existing CKAN fetcher.
- TPS: Toronto Police Open Data ArcGIS REST. Use `morie_fetch_tps(category = "Assault")`.
- SIU: Ontario SIU Director's Reports site. Use `morie_fetch_siu()` which parses the public reports site on demand (per-user, since redistribution of the parsed corpus is not clearly licensed).

Value

The on-demand fetchers (`morie_fetch_tps()`, `morie_fetch_siu()`) return the file path to the downloaded or cached CSV; `morie_load_dataset()` returns the loaded `data.frame`.

Examples

```
if (FALSE) {
  b01 <- morie_load_dataset("otisb01")
  head(b01)
}
```

`mrm_siu` *MRM-framework analyses on Ontario SIU (Special Investigations Unit) data*

Description

Three callables for SIU case-level CSVs. Unlike OTIS (no placement dates) and TPS (no per-person ID), SIU exposes per-case dates with a stable `police_service` jurisdiction column, enabling a real "time-to-outcome" KM survival analysis.

Details

Functions:

- `mrm_siu_case_to_decision_km()`: Kaplan-Meier on the gap from `date_of_incident_iso` to `date_of_director_decision_iso`, stratified by `police_service`. The valid TTR analysis the MA-thesis "210-day TTR" claim should have been.
- `mrm_siu_per_service_rate()`: Per-police-service case rate by year and stratum, useful for cross-jurisdiction comparisons.
- `mrm_siu_outcome_classifier()`: Tabulates the Director's-decision categories (`charges_laid`, `no_charges`, etc.) by service and by year, reporting both raw counts and shares.

Value

Each `mrm_siu_*()` callable returns a named `list` with the survival, per-service rate, or outcome-classification result and a plain-language `interpretation`.

Examples

```
if (FALSE) {
  siu <- read.csv("SIU.csv")
  mrm_siu_case_to_decision_km(siu)
}
```

`mrm_siu_case_to_decision_km`
KM survival of SIU case-to-decision gap per police service

Description

Computes the gap (in days) between the incident date and the Director's decision date for every SIU case, dropping rows where either date is missing. Reports per-stratum median + IQR + n. Cases without a decision date as of the snapshot are right-censored if `censor_open_cases = TRUE` (default).

Usage

```
mrm_siu_case_to_decision_km(
  data,
  incident_col = "date_of_incident_iso",
  decision_col = "date_of_director_decision_iso",
  service_col = "police_service",
  censor_open_cases = TRUE,
  min_n = 5L
)
```

Arguments

`data`	A data.frame with the SIU case schema.
`incident_col`	Column with ISO incident date (default `"date_of_incident_iso"`).
`decision_col`	Column with ISO Director's decision date (default `"date_of_director_decision_iso"`).
`service_col`	Stratifying jurisdiction column (default `"police_service"`).
`censor_open_cases`	Logical (default `TRUE`). If `TRUE`, rows with missing `decision_col` contribute right-censored observations from incident to the most recent decision date in the data set. If `FALSE`, they are dropped.
`min_n`	Minimum cases per service to retain in the per-service summary (default `5L`).

Details

This is the substantive "time-to-outcome" analysis the MA-thesis "210-day TTR" claim should have been; it operates on real per-case dates with a stable jurisdiction identifier.

Value

A list with elements:

- `pooled`: a single-row data.frame with the pooled median, mean, IQR, n, n_censored.
- `by_service`: per-service data.frame with the same columns.

Examples

```
if (FALSE) {
  siu <- read.csv("SIU.csv")
  res <- mrm_siu_case_to_decision_km(siu)
  head(res$by_service)
}
```

mrm_siu_outcome_classifier

Tabulate SIU Director's-decision outcomes

Description

Cross-tabulates a categorical outcome column (default `director_decision_category`) by service and year, reporting both raw counts and within-service shares. If the supplied `outcome_col` is not present, looks for a few common alternatives (`director_decision`, `outcome`).

Usage

```
mrm_siu_outcome_classifier(
  data,
  outcome_col = "director_decision_category",
  service_col = "police_service"
)
```

Arguments

`data`	A data.frame in the SIU case schema.
`outcome_col`	Outcome category column (default `"director_decision_category"`).
`service_col`	Police-service column (default `"police_service"`).

Value

A data.frame with columns `service`, `outcome`, `n_cases`, `share_within_service`.

Examples

```
if (FALSE) {
  siu <- read.csv("SIU.csv")
  mrm_siu_outcome_classifier(siu)
}
```

mrm_siu_per_service_rate

Per-police-service SIU case-rate summary

Description

Tabulates the number of SIU cases per police service per year, and optionally per `reason_for_interaction` stratum.

Usage

```
mrm_siu_per_service_rate(
  data,
  service_col = "police_service",
  incident_col = "date_of_incident_iso",
  stratify_col = NULL
)
```

Arguments

`data`	A data.frame in the SIU case schema.
`service_col`	Police-service column (default `"police_service"`).
`incident_col`	Incident-date column for year extraction (default `"date_of_incident_iso"`).
`stratify_col`	Optional second stratifying column.

Value

A data.frame with columns `service`, `year`, optional `stratum`, `n_cases`.

Examples

```
if (FALSE) {
  siu <- read.csv("SIU.csv")
  mrm_siu_per_service_rate(siu)
}
```

---

mrm_spatial_spillover

Spatial Durbin / SAR direct-indirect-total decomposition (MRM primitive)

Description

Mirrors the Python module `morie.mrm_primitives.spatial_spillover`, adapted from Laniyonu (2018) Urban Affairs Review 54(5):898-930, which in turn uses LeSage & Pace (2009) + Elhorst (2010) + the Yang/Noah/Shoff (2015) decomposition formula.

Details

The Laniyonu (2018) result – gentrification's effect on stops-per-capita is ~0 direct but +51 to +90\ into neighbouring tracts) – only surfaces once you decompose. An OLS or non-spatial FE model would report "no effect" and miss the entire story.

This primitive is the SDM with the canonical decomposition + the Moran's-I diagnostic that justifies SDM over OLS. We deliberately do NOT fit the SDM ourselves – **spdep/spatialreg** are hard deps we don't want to force. The caller passes the estimated ρ + β vectors; this primitive does the decomposition arithmetic.

`mrm_spatial_spillover_decomposition`
SDM direct / indirect / total decomposition

Description

Implements the standard LeSage & Pace formula:

Usage

```
mrm_spatial_spillover_decomposition(
  rho,
  beta_direct,
  beta_spatial,
  W,
  coefficient_names = NULL
)
```

Arguments

`rho`	Numeric scalar. Spatial-autoregressive coefficient from the fitted SDM.
`beta_direct`	Numeric vector of length K. Coefficients on the K covariates (no lagged terms).
`beta_spatial`	Numeric vector of length K. Coefficients on the spatially-lagged covariates (WX). Set to all zeros if you fit a SAR (lag-only) model.
`W`	Numeric matrix of shape (N, N). Row-standardised spatial weight matrix.
`coefficient_names`	Optional character vector of length K with human-readable covariate names; defaults to `c("x1", ..., "xK")`.

Details

$$(I - \rho W)^{-1}(I\beta_k + W\theta_k)$$

for each covariate k. The diagonal of the resulting per-observation effects matrix is averaged for the *direct* effect; the average off-diagonal-row-sum is the *indirect* effect; *total* = direct + indirect.

Value

A named list with classes `morie_mrm_result`, `morie_rich_result`, `list`. Carries `decomposition` (a `data.frame` with columns `coefficient`, `direct`, `indirect`, `total`, `note`), `rho`, plus `interpretation` + `warnings`.

Examples

```
set.seed(3)
N <- 12
W <- matrix(runif(N * N), N, N)
diag(W) <- 0
W <- W / rowSums(W)
res <- mrm_spatial_spillover_decomposition(
  rho = 0.4,
  beta_direct  = c(0.10, -0.05),
  beta_spatial = c(0.30,  0.00),
  W = W,
  coefficient_names = c("gentrification", "controls")
)
res$decomposition
```

---

mrm_standardised_difference

Imbens-Rubin standardised %SMD per covariate

Description

For continuous X: SMD = (mean_t - mean_c) / sqrt((s2_t + s2_c)/2) For binary X: SMD = (p_t - p_c) / sqrt((p_t(1-p_t) + p_c(1-p_c))/2) Returned as percent. |SMD| > 10 is the Austin (2009) imbalance threshold.

Usage

```
mrm_standardised_difference(data, treatment_col, covariates)
```

Arguments

`data`	data.frame.
`treatment_col`	Binary 0/1 treatment column name.
`covariates`	Character vector of covariate columns.

Value

data.frame with covariate, mean_treated, mean_control, pooled_sd, smd_pct, imbalanced columns.

Examples

```
set.seed(2026)
n <- 200L
df <- data.frame(
  D   = rbinom(n, 1, 0.4),
  age = rnorm(n, 50, 10),
  bmi = rnorm(n, 27, 4)
```

```
)
df$age[df$D == 1] <- df$age[df$D == 1] + 3 # deliberate imbalance
mrm_standardised_difference(df,
  treatment_col = "D",
  covariates = c("age", "bmi")
)
```

`mrm_synthetic_area_exposure` *Compute the synthetic exposure offset for each area*

Description

1. Fit logistic $P(\text{trait}|\text{covariates})$ on the survey microdata.
2. Apply fitted coefficients to area-level marginals from `area_df`.
3. Multiply predicted rate by area population to obtain the synthetic "population at risk" exposure offset.

Usage

```
mrm_synthetic_area_exposure(
  survey_df,
  survey_trait_col,
  survey_covariate_cols,
  area_df,
  area_population_col,
  fit_callable = NULL,
  return_per_area_rate = FALSE
)
```

Arguments

`survey_df`	A `data.frame` of survey microdata (one row per respondent), carrying `survey_trait_col` (0/1 or logical) and `survey_covariate_cols`.
`survey_trait_col`	Character. Name of the binary trait column.
`survey_covariate_cols`	Character vector of covariates that are present in BOTH the survey and the area dataset.
`area_df`	A `data.frame` with one row per area (tract, precinct, etc.); must carry the same covariate columns as area-level proportions / means, plus `area_population_col`.
`area_population_col`	Character. Adult-population column in `area_df`.
`fit_callable`	Optional function with signature `function(X, y) -> coef`, returning a coefficient vector of length `length(survey_covariate_cols) + 1L` (intercept first). Defaults to a base-R Newton-IRLS logistic fit.
`return_per_area_rate`	Logical; default `FALSE`. If `TRUE` the result list also carries `predicted_rate`.

Value

A named list with classes `morie_mrm_result`, `morie_rich_result`, `list`. Carries `exposure` (named numeric vector, one entry per area row), `predicted_rate` (when requested), `coef` (the fitted logistic coefficient vector), plus `interpretation` + `warnings`.

Examples

```
set.seed(2)
n_survey <- 500
x1 <- rnorm(n_survey); x2 <- rnorm(n_survey)
p  <- 1 / (1 + exp(-(-2 + 0.6 * x1 - 0.4 * x2)))
y  <- rbinom(n_survey, 1, p)
survey <- data.frame(trait = y, x1 = x1, x2 = x2)

area <- data.frame(
  x1 = rnorm(20), x2 = rnorm(20),
  pop = sample(800:1500, 20, replace = TRUE)
)
rownames(area) <- paste0("area_", seq_len(20))
res <- mrm_synthetic_area_exposure(
  survey_df = survey,
  survey_trait_col = "trait",
  survey_covariate_cols = c("x1", "x2"),
  area_df = area,
  area_population_col = "pop"
)
head(res$exposure)
```

`mrm_synthetic_exposure` *Synthetic small-area-estimated exposure offset (MRM primitive)*

Description

Mirrors the Python module `morie.mrm_primitives.synthetic_exposure`, adapted from Laniyonu & Goff (2021) BMC Psychiatry 21(1):500.

Details

The trick: when you need a rate-per-hidden-subpopulation (force- per-PwSMI, contact-per-undocumented, contact-per-homeless) and no administrative census of that subpopulation exists, you can:

1. Fit $P(\text{trait}|\text{covariates})$ on a national probability sample (NCS-R for SMI; ACS-style survey for other traits) using ONLY covariates also available at the area level.
2. Apply the fitted coefficients to area-level marginals from ACS / census to predict $P(\text{trait})$ per area.

3. Multiply by area-level adult population to get a synthetic "population at risk" denominator.

Generalises far beyond Laniyonu & Goff's SMI application: homelessness rates of police force, LGBTQ stop-and-frisk rates, undocumented-immigrant ICE-contact rates – any "rate per hidden subpopulation" estimand.

The returned offset is suitable for use as the `offset= log(exposure)` argument in a Poisson / negative-binomial GLM that counts trait-specific events.

`mrm_threshold_coefficient`
Extract coefficient(s) for one covariate across all thresholds

---

Description

Convenience accessor mirroring `ThresholdSpecificOrdinalResult.coefficient_by_threshold()`.

Usage

```
mrm_threshold_coefficient(x, covariate)
```

Arguments

`x`	A result from `mrm_threshold_specific_ordinal`.
`covariate`	Character, name of one covariate.

Value

A named numeric vector keyed by threshold label.

`mrm_threshold_specific_ordinal`
Fit a threshold-specific cumulative-logit ordinal regression

Description

For each cumulative cutpoint $k = 1, \ldots, K-1$, fits an independent logistic regression of $1\{Y \leq k\}$ on the covariates. Optionally fits the proportional-odds baseline and returns the likelihood-ratio test of PO vs. threshold-specific.

Usage

```
mrm_threshold_specific_ordinal(
  data,
  outcome_col,
  covariate_cols,
  ordinal_levels = NULL,
  fit_proportional_odds_first = TRUE,
  max_iter = 200L,
  tol = 1e-06
)
```

Arguments

`data`	data.frame, one row per unit.
`outcome_col`	Character; name of the ordinal outcome column. Either an ordered factor / integer code or a character column (in which case `ordinal_levels` should be passed explicitly).
`covariate_cols`	Character vector of predictor columns. Categorical predictors should be one-hot dummied before passing.
`ordinal_levels`	Optional character vector giving the explicit ordering of the outcome categories (low-to-high). If `NULL` and the outcome is a factor, `levels()` is used; otherwise `sort(unique())` (rarely what you want – pass this).
`fit_proportional_odds_first`	Logical; if `TRUE` (default) the proportional-odds baseline is fit and an LR test against the threshold-specific model is reported.
`max_iter`, `tol`	IRLS / GLM control passed to `glm.fit`.

Value

An object of class `c("mrm_threshold_specific_ordinal", "morie_mrm_result", "list")` with elements `threshold_labels`, `covariate_names`, `coefficients` (a (K-1) x p matrix), `cutpoints`, `log_likelihood`, `n_obs`, and (if requested) `proportional_odds_lr_stat`, `proportional_odds_lr_df` `proportional_odds_p`.

Examples

```
if (FALSE) {
  df <- data.frame(
    y = sample(c("low", "med", "high"), 200, replace = TRUE),
    race = rbinom(200, 1, 0.4),
    age  = rnorm(200)
  )
  mrm_threshold_specific_ordinal(df,
    outcome_col = "y",
    covariate_cols = c("race", "age"),
    ordinal_levels = c("low", "med", "high")
  )
```

```
}
```

`mrm_tps` *MRM-framework analyses on Toronto Police Service (TPS) open data*

Description

Four callables for TPS public-release crime-incident CSVs, used in the MRM empirical companion paper.

Details

Functions:

- `mrm_tps_levy_scaling()`: Hill-MLE Pareto exponent of inter-incident step-length distribution on the lat/long-coded event stream.
- `mrm_tps_moran_clustering()`: global Moran's I + DBSCAN cluster summary on the lat/long-coded event stream.
- `mrm_tps_neighbourhood_recurrence_km()`: Kaplan-Meier inter-event gap distribution per HOOD_158 neighbourhood.
- `mrm_tps_load_hawkes_refit()`: convenience loader that pulls the precomputed per-category Hawkes (Markovian + Weibull/sin) fits from the `paper_hawkes_refit.json` manifest if available.

Value

Each `mrm_tps_*()` callable returns a named `list` with the computed statistic (Pareto exponent, Moran's I, or survival curve) and a plain-language `interpretation`; `mrm_tps_load_hawkes_refit()` returns the parsed Hawkes-refit manifest as a `list`.

Examples

```
if (FALSE) {
  tps <- read.csv("Assault_Open_Data.csv")
  mrm_tps_levy_scaling(tps)
}
```

`mrm_tps_kulldorff_scan` *Run a 3-d (lat, lon, time) Kulldorff scan with MC inference*

Description

Run a 3-d (lat, lon, time) Kulldorff scan with MC inference

Usage

```
mrm_tps_kulldorff_scan(
  data,
  date_col = "OCC_DATE",
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84",
  radii_km = c(1, 2, 3, 5, 8),
  window_years = 4,
  n_centers = 60L,
  n_permutations = 199L,
  n_top_clusters = 1L,
  seed = 42L
)
```

Arguments

`data`	data.frame with date_col, lat_col, lon_col.
`date_col`	Column name of the event date (default `"OCC_DATE"`).
`lat_col`, `lon_col`	WGS84 lat/long column names.
`radii_km`	Candidate cylinder radii in km.
`window_years`	Time-cylinder length in years.
`n_centers`	Number of random candidate centres sub-sampled.
`n_permutations`	Monte-Carlo permutations.
`n_top_clusters`	Integer; number of top clusters to return. Accepted for Python signature parity. The current implementation returns a single primary cluster (the secondary-cluster loop in `morie.mrm_kulldorff.py` `break`s out pending a proper mask-and-rescan rewrite); values >1 are reserved for that future TRUE multi-cluster mode.
`seed`	Random seed.

Value

A one-row data.frame describing the top cluster, with columns `center_lat`, `center_lon`, `radius_km`, `t_start`, `t_end`, `n_observed`, `n_expected`, `relative_risk`, `log_lrt`, `p_value`.

Examples

```
if (FALSE) {
  tps <- morie_sample("tps_assault")
  mrm_tps_kulldorff_scan(tps, n_permutations = 49)
}
```

---

mrm_tps_levy_scaling *Levy-flight Hill-MLE exponent on TPS inter-incident step lengths*

Description

Treats consecutive events in chronological order as a single stream and computes the inter-event step length (km) via haversine on WGS84 latitude/longitude. Returns the Hill-MLE exponent restricted to steps above `min_step_km`.

Usage

```
mrm_tps_levy_scaling(
  data,
  date_col = "OCC_DATE",
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84",
  min_step_km = 0.5,
  x_min = NULL
)
```

Arguments

`data`	A data.frame with at least the columns named in `date_col`, `lat_col`, `lon_col`.
`date_col`	Column name of the date / timestamp (default `"OCC_DATE"`).
`lat_col`	Column name of WGS84 latitude (default `"LAT_WGS84"`).
`lon_col`	Column name of WGS84 longitude (default `"LONG_WGS84"`).
`min_step_km`	Lower-tail cutoff in km (default `0.5`).
`x_min`	Hill-MLE cutoff (default = `min_step_km`).

Value

A list with `n_events`, `n_steps_tail`, `min_step_km`, `hill_alpha`.

Examples

```
if (FALSE) {
  tps <- read.csv("Assault_Open_Data.csv")
  mrm_tps_levy_scaling(tps)
}
```

`mrm_tps_lisa` *Local Moran's I per polygon + quadrant + 999-permutation significance*

Description

Local Moran's I per polygon + quadrant + 999-permutation significance

Usage

```
mrm_tps_lisa(
  data,
  count_col,
  lat_col = "lat",
  lon_col = "lon",
  id_col = NULL,
  k = 6L,
  n_permutations = 999L,
  seed = 42L
)
```

Arguments

`data`	data.frame with one row per polygon.
`count_col`	Column with per-polygon counts (e.g. "ASSAULT_2024").
`lat_col`, `lon_col`	WGS84 centroid columns.
`id_col`	Optional polygon-ID column (passed through to output).
`k`	k-NN spatial-weights neighbourhood (default 6).
`n_permutations`	MC permutations (default 999, the spatial-statistics convention).
`seed`	RNG seed.

Value

A list with elements `n_polygons`, `global_moran_I`, `permutations`, `knn_k`, `table` (per-polygon data.frame), `quadrants_all`, `quadrants_significant_p05`, `n_significant_p05`.

Examples

```
if (FALSE) {
  ncr <- read.csv("Neighbourhood_Crime_Rates_Open_Data.csv")
  res <- mrm_tps_lisa(ncr,
    count_col = "ASSAULT_2024",
    lat_col = "lat", lon_col = "lon"
  )
}
```

`mrm_tps_load_hawkes_refit` *Load the precomputed per-category TPS Hawkes refit manifest*

Description

Reads `paper_hawkes_refit.json` and returns the Markovian (exponential kernel, constant baseline) and non-Markovian (Weibull kernel, sinusoidal baseline) AIC, branching ratio, and KS p-value per category as a tidy data.frame.

Usage

```
mrm_tps_load_hawkes_refit(manifest_path)
```

Arguments

`manifest_path` Path to `paper_hawkes_refit.json`.

Value

A data.frame with one row per category, columns `category`, `n_fitted`, `T_days`, `aic_mark`, `kappa_mark`, `ks_p_mark`, `aic_nm`, `eta_nm`, `ks_p_nm`, `delta_aic`.

Examples

```
if (FALSE) {
  mrm_tps_load_hawkes_refit("paper_hawkes_refit.json")
}
```

`mrm_tps_moran_clustering` *Global Moran's I + DBSCAN summary on TPS lat/long event data*

Description

Grids the lat/long extent of `data` into a coarse raster of `grid_resolution` cells, counts events per cell, and computes the global Moran's I via a rook contiguity matrix. Also runs DBSCAN on the raw lat/long points (rescaled to km) and reports cluster counts.

Usage

```
mrm_tps_moran_clustering(
  data,
  lat_col = "LAT_WGS84",
  lon_col = "LONG_WGS84",
  grid_resolution = 40L,
  dbscan_eps = 0.3,
  dbscan_minpts = 5L
)
```

Arguments

`data`	A data.frame with `lat_col` and `lon_col`.
`lat_col`	Column name of WGS84 latitude.
`lon_col`	Column name of WGS84 longitude.
`grid_resolution`	Number of cells per axis (default `40L`).
`dbscan_eps`	DBSCAN radius in km (default `0.3`).
`dbscan_minpts`	DBSCAN minimum points per core (default `5L`).

Details

This function is a thin computational wrapper. For high-precision computations on full-sized TPS files use the `morie` Python `tps_spatial_advanced` pipeline; the R version is for quick interactive auditing.

Value

A list with `morans_I`, `morans_z`, `dbscan_n_clusters`, `dbscan_n_noise`, `dbscan_largest`.

Examples

```
if (FALSE) {
  tps <- read.csv("Assault_Open_Data.csv")
  mrm_tps_moran_clustering(tps)
}
```

---

mrm_tps_neighbourhood_recurrence_km

Kaplan-Meier inter-event recurrence on TPS by neighbourhood

Description

For each HOOD_158 neighbourhood, sorts events chronologically and computes the gap (in days) between consecutive events. Returns the per-neighbourhood mean, median, and total gaps. No censoring is applied (every gap is observed).

Usage

```
mrm_tps_neighbourhood_recurrence_km(
  data,
  date_col = "OCC_DATE",
  hood_col = "HOOD_158",
  min_gap_days = 0
)
```

Arguments

`data`	A data.frame with `date_col` and `hood_col`.
`date_col`	Column name of the date column (default `"OCC_DATE"`).
`hood_col`	Column name of the neighbourhood ID (default `"HOOD_158"`).
`min_gap_days`	Smallest gap to include (default `0`).

Value

A data.frame with one row per neighbourhood, columns `hood`, `n_events`, `n_gaps`, `mean_gap_days`, `median_gap_days`, `p25_gap_days`, `p75_gap_days`.

Examples

```
if (FALSE) {
  tps <- read.csv("Assault_Open_Data.csv")
  mrm_tps_neighbourhood_recurrence_km(tps)
}
```

---

mrm_tps_polygon_moran_per_year

Per-year global Moran's I time series across a polygon surface

---

Description

Convenience wrapper that loops `mrm_tps_lisa` over a vector of per-year count columns.

Usage

```
mrm_tps_polygon_moran_per_year(
  data,
  year_cols,
  lat_col = "lat",
  lon_col = "lon",
  k = 6L,
  n_permutations = 999L,
  seed = 42L
)
```

Arguments

`data`	Polygon-level data.frame.
`year_cols`	Character vector of per-year count column names (e.g. `c("ASSAULT_2014", ..., "ASSAULT_2024")`).
`lat_col`, `lon_col`, `k`, `n_permutations`, `seed`	as in `mrm_tps_lisa`.

Value

data.frame with columns `year`, `n_events`, `moran_I`, `global_p_value`.

Examples

```
# 4 x 4 polygon grid with two yearly count columns.
set.seed(2026)
grid <- expand.grid(
  lat = 43.6 + (0:3) * 0.02,
  lon = -79.4 + (0:3) * 0.02
)
grid$ASSAULT_2023 <- rpois(nrow(grid), lambda = grid$lat * 10)
grid$ASSAULT_2024 <- rpois(nrow(grid), lambda = grid$lat * 12)
res <- mrm_tps_polygon_moran_per_year(
  grid,
  year_cols = c("ASSAULT_2023", "ASSAULT_2024"),
  lat_col = "lat", lon_col = "lon",
  k = 4L, n_permutations = 99L, seed = 42L
)
res
```

`mrm_two_treatment_test` *Two-treatment outcome comparison with three assumption regimes*

Description

Always returns Welch t (unequal variance, canonical), Student t (equal variance), and Mann-Whitney U (rank-based). The Welch p-value is the canonical answer; the others are the sensitivity range.

Usage

```
mrm_two_treatment_test(a, b, alpha = 0.05)
```

Arguments

`a`, `b`	Outcome vectors under treatments A and B.
`alpha`	CI level (default 0.05).

Value

Named list with estimate, se, t_statistic, df, p_welch, p_student, p_mannwhitney, ci_lower, ci_upper, n_a, n_b, interpretation.

Examples

```
set.seed(2026)
a <- rnorm(40, mean = 5, sd = 1.2)
b <- rnorm(40, mean = 5.5, sd = 1.5)
res <- mrm_two_treatment_test(a, b)
res$estimate # mean(a) - mean(b)
res$p_welch # canonical p-value
res$p_mannwhitney # rank-based sensitivity check
```

---

mrm_twoprop_test *Two-proportion test (chi-square + Fisher exact + Wald)*

Description

Two-proportion test (chi-square + Fisher exact + Wald)

Usage

```
mrm_twoprop_test(x1, n1, x2, n2, alpha = 0.05)
```

Arguments

x1, n1	Successes and trials in group 1.
x2, n2	Successes and trials in group 2.
alpha	CI level (default 0.05).

Value

Named list with p1, p2, diff, chi2, df, p_value_chi2, p_value_fisher, z_wald, p_value_wald, ci95_diff_lower/upper, interpretation.

Examples

```
# Compare 47/100 vs 31/100; two-sided test.
mrm_twoprop_test(x1 = 47, n1 = 100, x2 = 31, n2 = 100)
```

`mrm_uof` *Generic Multilevel Reconciliation Methodology (MRM) Use-of-Force callables*

Description

Six jurisdiction-agnostic analyses for police Use-of-Force data, mirroring the Python module `morie.mrm_uof`. Every function accepts a `data.frame` (or `tibble`) and returns a named `list` carrying both the numeric outputs and a multi-paragraph plain-language `interpretation`, so the result can be printed to a notebook without further post-processing.

Details

Functions:

- `mrm_uof_force_concentration`: Hill-MLE Pareto exponent + Gini coefficient + top-5 / top-10 share for incident counts aggregated by force / service.
- `mrm_uof_weapon_diversity`: weapon-by-force contingency: chi-square, Cramer's V, and the top-3 cells by standardised Pearson residual.
- `mrm_uof_yoy_change`: year-on-year percentage change with a manual largest-gap change-point fallback (the R side does not require **ruptures**).
- `mrm_uof_region_locality`: region-at-time vs. region-now contingency: diagonal share, chi-square, Cramer's V.
- `mrm_uof_demographic_disparity`: per-category outcome rates with Wilson 95\ baseline group, optional non-parametric bootstrap percentile interval on the risk ratio.
- `mrm_uof_data_quality_audit`: per-column null and dtype audit, with optional schema-comparison against a supplied CKAN sidecar list or column-spec list.

`mrm_uof_data_quality_audit` *Schema, null, and suspect-value audit*

Description

Schema, null, and suspect-value audit

Usage

```
mrm_uof_data_quality_audit(df, sidecar = NULL, expected_schema = NULL)
```

Arguments

`df`	A data.frame.
`sidecar`	Optional list with `fields` (list-of-list with `id` and `type`) and optionally `records`, in the CKAN `datastore_search` response shape.
`expected_schema`	Optional list with a `columns` field carrying named entries with `name` and `dtype`.

Value

Named list with `per_column`, `missing_columns`, `extra_columns`, `dtype_mismatches`, `suspect_flags`.

`mrm_uof_demographic_disparity`
Demographic disparity in outcome rates with risk-ratio CIs

Description

Demographic disparity in outcome rates with risk-ratio CIs

Usage

```
mrm_uof_demographic_disparity(
  df,
  demo_col,
  outcome_col,
  baseline = NULL,
  bootstrap_reps = 0L
)
```

Arguments

`df`	A data.frame.
`demo_col`	Categorical demographic column.
`outcome_col`	Binary outcome column (0/1 or logical).
`baseline`	Optional baseline category (default: largest-N group).
`bootstrap_reps`	Bootstrap replications for the RR percentile CI. Set to 0 (default) to skip.

Value

Named list with `baseline`, `baseline_rate`, `per_category` (list of lists), `risk_ratios`.

`mrm_uof_force_concentration`
Concentration of UoF incidents across forces / services

Description

Aggregates per-force incident counts and reports a Hill-MLE Pareto tail exponent, the Gini coefficient, and the top-5 / top-10 concentration shares.

Usage

```
mrm_uof_force_concentration(df, force_col, count_col = NULL)
```

Arguments

df	A data.frame or tibble with one row per incident (when count_col is NULL) or one row per force-period with a numeric count_col.
force_col	Character. Name of the column identifying the force / service / agency.
count_col	Character or NULL. If supplied, the per-row incident count to sum within each force; otherwise each row counts as one incident.

Value

A named list with classes morie_mrm_uof_result, morie_rich_result, list. Numeric outputs include pareto_alpha_mle, gini, top5_share, top10_share, n_forces, n_incidents.

Examples

```
df <- data.frame(force = c(rep("A", 50), rep("B", 5)))
res <- mrm_uof_force_concentration(df, "force")
res$gini
```

---

mrm_uof_region_locality

Region-at-time vs region-now locality contingency

Description

Region-at-time vs region-now locality contingency

Usage

```
mrm_uof_region_locality(df, region_at_col, region_now_col)
```

Arguments

df	A data.frame.
region_at_col	Region at the time of the incident.
region_now_col	Most-recent region.

Value

Named list with diagonal_share, chi2, pvalue, df, cramers_v.

mrm_uof_weapon_diversity

Weapon-by-force contingency test

Description

Builds a weapon x force contingency table, runs a chi-square test of independence, computes Cramer's V, and reports the top-3 (weapon, force) cells by standardised Pearson residual.

Usage

```
mrm_uof_weapon_diversity(df, weapon_col, force_col)
```

Arguments

`df`	A data.frame.
`weapon_col`	Categorical weapon column.
`force_col`	Categorical force / service column.

Value

A named list with `chi2`, `pvalue`, `df`, `cramers_v`, `top_residuals` (list-of-lists), and an `interpretation` paragraph.

mrm_uof_yoy_change

Year-on-year change in incident counts

Description

Either supply `dfs_by_year` (named list mapping year string / integer to a data.frame) or `df` + `year_col`.

Usage

```
mrm_uof_yoy_change(
  dfs_by_year = NULL,
  df = NULL,
  year_col = NULL,
  count_col = NULL
)
```

Arguments

`dfs_by_year`	Named list of `data.frame`s, names coerced to integer years.
`df`	A data.frame to be grouped by `year_col`.
`year_col`	Required when `df` is supplied.
`count_col`	Optional column to sum within each year (rows counted otherwise).

Details

Change-point detection is the manual largest-absolute-difference heuristic (the R port does not require **changepoint**).

Value

Named list with `years`, `counts`, `yoy_pct`, `change_point_year`, `mean_abs_yoy_pct`.

`mrm_var_test` *Chi-square test for variance (Wilks 1962)*

Description

Chi-square test for variance (Wilks 1962)

Usage

```
mrm_var_test(sample, sigma0_sq, alpha = 0.05)
```

Arguments

`sample`	Numeric vector (assumed iid normal).
`sigma0_sq`	Null hypothesis variance.
`alpha`	CI level (default 0.05).

Value

Named list with s_sq, sigma0_sq, chi2_stat, df, p_value_two_sided, p_value_one_sided_greater/less, ci95_lower/upper, interpretation.

Examples

```
set.seed(2026)
x <- rnorm(50, mean = 0, sd = 1.2)
# H0: variance = 1.
mrm_var_test(sample = x, sigma0_sq = 1)
```

n_effective_tests *Effective number of independent tests from a correlation matrix*

Description

Delegates to `poolr::meff` when installed (`poolr` implements Galwey, Li-Ji, and Nyholt). Otherwise computes the chosen estimator inline.

Usage

```
n_effective_tests(correlation_matrix, method = c("galwey", "li_ji", "nyholt"))
```

Arguments

`correlation_matrix`	Square symmetric correlation matrix.
`method`	One of `"galwey"` (Galwey 2009), `"li_ji"` (Li and Ji 2005), or `"nyholt"` (Nyholt 2004).

Value

Effective number of tests (>= 1).

n_stat_commands *Total number of registered commands (excluding aliases)*

Description

Total number of registered commands (excluding aliases)

Usage

```
n_stat_commands()
```

`nested_cross_validate` *Nested cross-validation with inner-loop grid search*

Description

Performs nested K-fold CV: the outer loop estimates generalisation performance while an inner CV grid search picks the best hyperparameter configuration on each outer training fold. Two calling conventions are supported for backward compatibility:

Usage

```
nested_cross_validate(
  fit_fn = NULL,
  predict_fn = NULL,
  X = NULL,
  y = NULL,
  score_fn = NULL,
  hyperparam_grid = NULL,
  outer_k = 5L,
  inner_k = 3L,
  scoring = "roc_auc",
  random_state = 42L,
  tune_fn = NULL,
  outer_folds = NULL
)
```

Arguments

`fit_fn`	Function with signature `(X, y, hyperparams) -> model` accepting a single hyperparameter list (full form only).
`predict_fn`	Function with signature `(model, X) -> y_pred`.
`X`	Numeric predictor matrix (or coercible).
`y`	Response vector.
`score_fn`	Optional custom scoring function `(y_true, y_pred) -> numeric(1)`. Higher is better. If `NULL`, the named scoring rule via `scoring` is used.
`hyperparam_grid`	Named list of candidate vectors (one per hyperparameter). The Cartesian product defines the search grid.
`outer_k`	Number of outer folds (default 5).
`inner_k`	Number of inner folds (default 3).
`scoring`	Named scoring rule passed to the internal scorer (`"roc_auc"`, `"accuracy"`, `"brier"`). Used only if `score_fn` is `NULL`.
`random_state`	Integer seed for fold construction (default 42).
`tune_fn`	Deprecated legacy positional argument; see Description.
`outer_folds`	Deprecated alias for `outer_k` (legacy stub form).

Details

- **Legacy stub form:** `nested_cross_validate(tune_fn, predict_fn, X, y, outer_folds, scoring, random_state)` where `tune_fn(X, y)` returns a fitted model (no grid argument). In this mode no inner search is run.
- **Full form:** pass `fit_fn`, `predict_fn`, `score_fn`, and `hyperparam_grid` (a named list of candidate vectors). The function enumerates the Cartesian product, runs inner K-fold CV on each outer training fold, picks the best configuration, refits on the full outer-train fold, and scores on the held-out outer fold.

Value

Named list with `outer_scores` (numeric vector, length `outer_k`), `best_hyperparams_per_fold` (list of named lists), `mean_score`, `se_score`, and `n_configs`.

Examples

```
set.seed(1)
n <- 120
X <- matrix(rnorm(n * 3), n, 3)
y <- as.integer(plogis(X[, 1]) > runif(n))
fit_fn <- function(X, y, hp) {
  df <- data.frame(y = y, X)
  suppressWarnings(stats::glm(y ~ ., data = df, family = stats::binomial()))
}
predict_fn <- function(model, X) {
  stats::predict(model, newdata = data.frame(X), type = "response")
}
nested_cross_validate(fit_fn = fit_fn, predict_fn = predict_fn,
                      X = X, y = y,
                      hyperparam_grid = list(dummy = c(1)),
                      outer_k = 3L, inner_k = 2L)
```

`normality_suite` *Run a suite of normality tests*

Description

Internal Shapiro-Wilk and Jarque-Bera helpers were removed in v0.9.6; the suite now calls `stats::shapiro.test` directly and computes the Jarque-Bera statistic inline. Users wanting a stand-alone Jarque-Bera test should call `tseries::jarque.bera.test`.

Usage

```
normality_suite(x)
```

Arguments

`x`	Numeric vector.

Value

A list of `morie_test_result`.

`number_needed_to_harm` *Number needed to harm (NNH) — sign-reversed NNT*

Description

Number needed to harm (NNH) — sign-reversed NNT

Usage

```
number_needed_to_harm(a, b, c, d, confidence = 0.95)
```

Arguments

`a`, `b`, `c`, `d`	Cell counts.
`confidence`	Confidence level. Default 0.95.

Value

A `morie_effect_size`.

`number_needed_to_treat` *Number needed to treat (NNT) = 1 / |RD|*

Description

Number needed to treat (NNT) = 1 / |RD|

Usage

```
number_needed_to_treat(a, b, c, d, confidence = 0.95)
```

Arguments

`a`, `b`, `c`, `d`	Cell counts.
`confidence`	Confidence level. Default 0.95.

Value

A `morie_effect_size`.

nw_regression — *Nadaraya-Watson kernel regression*

Description

Computes the kernel-weighted local mean estimator m-hat(x) = sum K_h(x - X_i) Y_i divided by sum K_h(x - X_i), with a Gaussian kernel.

Usage

```
nw_regression(x, y, x_eval, bandwidth)
```

Arguments

`x`	Numeric vector of observed covariate values, length n.
`y`	Numeric vector of observed outcomes, length n.
`x_eval`	Numeric vector of evaluation points.
`bandwidth`	Positive bandwidth h.

Value

Numeric vector of fitted values at `x_eval`.

References

Nadaraya, E. A. (1964). On Estimating Regression. *Theory of Probability and Its Applications*, 9(1), 141-142.

odds_ratio — *Odds ratio for a 2x2 table* `[[a, b], [c, d]]`

Description

Odds ratio for a 2x2 table `[[a, b], [c, d]]`

Usage

```
odds_ratio(a, b, c, d, confidence = 0.95)
```

Arguments

`a`, `b`, `c`, `d`	Cell counts.
`confidence`	Confidence level. Default 0.95.

Value

A `morie_effect_size`.

odds_ratio_table *Odds-ratio table from a fitted logistic GLM*

Description

Odds-ratio table from a fitted logistic GLM

Usage

```
odds_ratio_table(
  model,
  confidence = 0.95,
  digits = 3L,
  apa = FALSE,
  output_format = "dataframe",
  title = "Odds Ratios"
)
```

Arguments

model	A fitted glm with family = binomial().
confidence	Confidence level.
digits	Decimal places.
apa	APA formatting.
output_format	Output target.
title	Title.

omega_squared *Omega-squared — less biased than eta-squared*

Description

Omega-squared — less biased than eta-squared

Usage

```
omega_squared(ss_effect, ss_total, df_effect, ms_error)
```

Arguments

ss_effect, ss_total	Sums of squares.
df_effect	Numerator d.f. of the effect.
ms_error	Error mean square.

Value

A `morie_effect_size`.

`omitted_variable_bias` *Omitted-variable bias analysis (sensemakr framework)*

Description

Closed-form Cinelli-Hazlett robustness-value implementation in base R. For the full **sensemakr** treatment (benchmark plots, adjusted t-statistics, contour plots) on a fitted `lm` object, use `morie_sensitivity_omitted_var_bias`.

Usage

```
omitted_variable_bias(
  estimate,
  se,
  dof,
  r2_yd_x,
  partial_r2_treatment,
  q = 1,
  alpha = 0.05,
  benchmark_covariates = NULL
)
```

Arguments

`estimate`	Treatment coefficient.
`se`	SE of the estimate.
`dof`	Residual degrees of freedom.
`r2_yd_x`	Partial R^2 of treatment with outcome.
`partial_r2_treatment`	Same as `r2_yd_x` (for clarity).
`q`	Fraction of the estimate to be explained away. Default 1.
`alpha`	Significance level. Default 0.05.
`benchmark_covariates`	Named list mapping covariate name -> partial R^2.

Value

A `morie_ovb` named-list.

`one_proportion_ztest` *One-sample z-test for a proportion*

Description

One-sample z-test for a proportion

Usage

```
one_proportion_ztest(count, nobs, value = 0.5, confidence = 0.95)
```

Arguments

`count`	Successes.
`nobs`	Total observations.
`value`	Hypothesised proportion.
`confidence`	Confidence level (Wilson CI).

`one_sample_ttest` *One-sample Student's t-test*

Description

One-sample Student's t-test

Usage

```
one_sample_ttest(x, mu0 = 0, confidence = 0.95)
```

Arguments

`x`	Numeric vector.
`mu0`	Hypothesised mean.
`confidence`	Confidence level (default 0.95).

Value

A `morie_test_result`.

`one_way_anova` *One-way between-subjects ANOVA*

Description

One-way between-subjects ANOVA

Usage

```
one_way_anova(...)
```

Arguments

`...`	Two or more numeric vectors (groups).

Value

`morie_test_result` with eta-squared effect size.

`or_to_d` *Convert odds ratio to Cohen's d (Hasselblad & Hedges, 1995)*

Description

Convert odds ratio to Cohen's d (Hasselblad & Hedges, 1995)

Usage

```
or_to_d(or_val)
```

Arguments

`or_val`	Odds ratio.

Value

Numeric d.

or_to_r *Convert OR to Pearson r via d*

Description

Convert OR to Pearson r via d

Usage

```
or_to_r(or_val)
```

Arguments

or_val	Odds ratio.

Value

Numeric r.

paired_permutation_test *Paired permutation test (sign-flipping)*

Description

Performs a sign-flipping paired permutation test on the paired differences. `coin::symmetry_test(distribution = "approximate")` is the canonical CRAN equivalent (cross-referenced); rmorie keeps the inline path because the rmorie API returns the full null distribution.

Usage

```
paired_permutation_test(
  x,
  y,
  statistic = "mean_diff",
  n_permutations = 9999L,
  alternative = "two-sided",
  seed = 42L
)
```

Arguments

x, y	Paired numeric vectors (same length).
statistic	`"mean_diff"` or `"median_diff"`.
n_permutations	Number of permutations.
alternative	`"two-sided"`, `"greater"`, `"less"`.
seed	Random seed.

Value

A `morie_permutation_test_result`.

See Also

`coin::symmetry_test`.

---

paired_ttest *Paired-sample t-test*

Description

Paired-sample t-test

Usage

```
paired_ttest(x, y, confidence = 0.95)
```

Arguments

`x, y`	Equal-length numeric vectors.
`confidence`	Confidence level.

---

parametric_bootstrap *Parametric bootstrap*

Description

Generates bootstrap samples from a fitted parametric distribution rather than from the empirical sample. Delegates to `boot::boot(sim = "parametric")` when **boot** is installed; otherwise uses an inline `rnorm/rpois/rbinom/rexp/rgamma` loop.

Usage

```
parametric_bootstrap(
  data,
  statistic,
  distribution = "normal",
  n_boot = 2000L,
  ci_level = 0.95,
  seed = 42L,
  ...
)
```

Arguments

`data`	Original numeric data (used to fit the distribution).
`statistic`	Function returning a scalar.
`distribution`	One of `"normal"`, `"poisson"`, `"binomial"`, `"exponential"`, `"gamma"`.
`n_boot`	Number of replicates.
`ci_level`	Confidence level.
`seed`	Random seed.
`...`	Distribution-specific parameters (mu, sigma, lam, p, scale, shape).

Value

A `morie_bootstrap_result`.

See Also

`boot::boot`.

`partial_correlation` *Partial Pearson correlation controlling for covariates*

Description

Partial Pearson correlation controlling for covariates

Usage

```
partial_correlation(x, y, covariates, confidence = 0.95)
```

Arguments

`x, y`	Numeric vectors of interest.
`covariates`	Matrix or data frame of covariates.
`confidence`	Confidence level.

`partial_eta_squared` *Partial eta-squared*

Description

Partial eta-squared

Usage

```
partial_eta_squared(ss_effect, ss_error)
```

Arguments

`ss_effect`	Sum of squares for the effect.
`ss_error`	Error sum of squares.

Value

A `morie_effect_size`.

`pburg` *Burg autoregressive power spectral density*

Description

Burg AR-spectrum estimation: parametric PSD via the Burg algorithm for AR-coefficient estimation. Well-suited to short HRV windows where Welch suffers from low spectral resolution.

Usage

```
pburg(x, fs, order = 16L, nfft = 256L)
```

Arguments

`x`	Numeric vector (1-D signal).
`fs`	Sampling frequency in Hz.
`order`	AR model order (default 16).
`nfft`	FFT length for PSD evaluation (default 256).

Details

Reference: Marple, S.L. (1987) *Digital Spectral Analysis with Applications*, Prentice-Hall, on the Burg algorithm.

Value

List with `filtered` (PSD), `name`, `fs`, `n_samples`, and `extra` (`freqs`, `order`, `ar_coefficients`).

Examples

```
set.seed(1)
t <- seq(0, 1, length.out = 512)
x <- sin(2 * pi * 10 * t) + 0.5 * rnorm(length(t))
res <- pburg(x, fs = 512)
length(res$filtered)
```

---

pcgenv *PCG Shannon-energy envelope*

Description

Shannon-energy envelope of a phonocardiogram (PCG): normalises the signal, computes $-x^2 \log x^2$, then box-smooths over a 20 ms window. The standard envelope used for S1/S2 segmentation.

Usage

```
pcgenv(pcg, fs)
```

Arguments

`pcg`	Numeric vector (1-D PCG signal).
`fs`	Sampling frequency in Hz.

Details

Reference: Liang, H., Lukkarinen, S. & Hartimo, I. (1997) "Heart sound segmentation algorithm based on heart sound envelogram", *Comput. Cardiol.*, pp. 105–108.

Value

List with `filtered` (envelope), `name`, `fs`, `n_samples`.

Examples

```
set.seed(1)
pcg <- rnorm(4000)
res <- pcgenv(pcg, fs = 2000)
length(res$filtered)
```

pcgmur *PCG murmur likelihood score*

Description

Combines a 100–400 Hz band-energy ratio, normalised spectral entropy, and the Higuchi fractal dimension of the PCG into a murmur-likelihood score in `[0, 1]`.

Usage

```
pcgmur(pcg, fs)
```

Arguments

`pcg`	Numeric vector (1-D PCG signal).
`fs`	Sampling frequency in Hz.

Details

Reference: Rangayyan, R.M. (2015) *Biomedical Signal Analysis*, 2nd ed., Wiley/IEEE Press, chapter on heart-sound analysis.

Value

List with `value` (score in `[0, 1]`), `name`, and `extra` (`fractal_dimension`, `hf_energy_ratio`, `spectral_entropy`, `fd_score`, `hf_score`, `ent_score`).

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  pcg <- rnorm(4000)
  res <- pcgmur(pcg, fs = 2000)
  res$value
}
```

pcgseg *PCG S1/S2 heart-sound segmentation*

Description

Segments a PCG Shannon-energy envelope into S1 (systolic) and S2 (diastolic) heart-sound events: threshold, find above-threshold runs, merge close peaks, label alternating events.

Usage

```
pcgseg(envelope, fs = 2000, min_gap_ms = 100)
```

Arguments

`envelope`	Numeric vector (Shannon-energy envelope).
`fs`	Sampling frequency in Hz (default 2000).
`min_gap_ms`	Minimum gap between peaks in ms (default 100).

Details

Reference: Liang, Lukkarinen & Hartimo (1997), *Comput. Cardiol.*, pp. 105–108.

Value

List with `value` (cycle count), `name`, and `extra` (`s1_indices`, `s2_indices`, `n_cycles`, `n_peaks`).

Examples

```
set.seed(1)
env <- abs(sin(seq(0, 20, length.out = 4000))) + 0.05 * rnorm(4000)
env[env < 0] <- 0
res <- pcgseg(env, fs = 2000)
res$extra$n_cycles
```

`pearson_correlation` *Pearson product-moment correlation*

Description

Pearson product-moment correlation

Usage

```
pearson_correlation(x, y, confidence = 0.95)
```

Arguments

`x`, `y`	Numeric vectors.
`confidence`	Confidence level.

`permutation_fdr` *Permutation-based FDR control via empirical null p-value distribution*

Description

Estimates the false discovery rate at each candidate threshold using a matrix of p-values computed under the permutation null, and selects the largest threshold whose estimated FDR is at most `alpha`. Q-values are assigned as the minimum estimated FDR across thresholds at least as large as each observed p-value.

Usage

```
permutation_fdr(test_stats, null_stats, alpha = 0.05, labels = NULL)
```

Arguments

`test_stats`	Numeric vector of observed p-values (named after the Python sibling argument to keep cross-language parity).
`null_stats`	Numeric matrix of permutation-null p-values with n_{perm} rows and m columns.
`alpha`	Target FDR level (default 0.05).
`labels`	Optional character vector of test labels.

Value

A `morie_rich_result` list with `original` (raw p-values), `adjusted` (q-values), `rejected`, `method`, `alpha`, `n_rejected`, `n_tests`.

Examples

```
set.seed(1)
m <- 20; nperm <- 200
p_obs <- c(stats::runif(m - 4), c(1e-4, 1e-3, 1e-3, 5e-3))
p_null <- matrix(stats::runif(nperm * m), nperm, m)
permutation_fdr(p_obs, p_null)
```

`permutation_fwer` *Permutation-based FWER control via step-down max-T (Westfall-Young)*

Description

Given observed test statistics and a matrix of test statistics under the permutation null, computes step-down max-T adjusted p-values that strongly control the family-wise error rate without requiring independence across tests.

Usage

```
permutation_fwer(
  test_stats,
  null_stats,
  alternative = c("two_sided", "greater", "less"),
  alpha = 0.05,
  labels = NULL
)
```

Arguments

`test_stats`	Numeric vector of length m of observed test statistics.
`null_stats`	Numeric matrix with n_{perm} rows and m columns containing test statistics computed on permuted data.
`alternative`	One of `"two_sided"` (default), `"greater"`, or `"less"`.
`alpha`	Significance level for rejection (default 0.05).
`labels`	Optional character vector of test labels.

Value

A `morie_rich_result` list with `original`, `adjusted`, `rejected`, `method`, `alpha`, `n_rejected`, `n_tests`.

Examples

```
set.seed(1)
m <- 10; nperm <- 200
obs <- c(rnorm(m - 2), 4.0, 3.5)
null <- matrix(rnorm(nperm * m), nperm, m)
permutation_fwer(obs, null)
```

`permutation_test` *Two-sample permutation test*

Description

Shuffles the combined samples `n_permutations` times to construct the null distribution of the chosen test statistic. **coin**'s `coin::oneway_test(distribution = "approximate")` implements the same test with a Monte-Carlo null; it is delegated to when **coin** is installed and `statistic` is the default `"mean_diff"` (the rmorie API allows arbitrary `f(g1, g2)` which coin does not expose, so a custom statistic falls back to the inline shuffle loop). The inline path keeps the full null distribution which downstream MRM code consumes.

Usage

```
permutation_test(
  group1,
  group2,
  statistic = "mean_diff",
  n_permutations = 9999L,
  alternative = "two-sided",
  seed = 42L
)
```

Arguments

`group1`, `group2`	Numeric vectors.
`statistic`	Either `"mean_diff"`, `"median_diff"`, `"t_stat"`, or a function `f(g1, g2) -> scalar`.
`n_permutations`	Number of permutations.
`alternative`	`"two-sided"`, `"greater"`, or `"less"`.
`seed`	Random seed.

Value

A `morie_permutation_test_result`.

See Also

`coin::oneway_test`, `coin::independence_test`.

`pettitt_changepoint` *Pettitt non-parametric change-point detection*

Description

Pettitt's (1979) test for a single change-point in a time series. Returns the change-point index, the test statistic, and an approximate p-value.

Usage

```
pettitt_changepoint(series)
```

Arguments

`series`	Numeric vector / time series.

Value

Named list with `change_point_index`, `U_max`, `p_value`, `note`.

References

Pettitt (1979). A non-parametric approach to the change-point problem. *J. R. Stat. Soc. C*, 28(2), 126–135.

`pfd` *Petrosian fractal dimension*

Description

Petrosian fractal dimension $D =$
$log_{10}(N)/($
$log_{10}(N) +$
$log_{10}(N/(N + 0.4N_{\delta})))$, where N_{δ} counts sign changes of the first difference. A fast complexity proxy for EEG/ECG.

Usage

```
pfd(x)
```

Arguments

`x`	Numeric vector.

Details

Reference: Petrosian, A. (1995) "Kolmogorov complexity of finite sequences and recognition of different preictal EEG patterns", *Proc. 8th IEEE Symp. Comput.-Based Med. Syst.*, pp. 212–217.

Value

List with `value` (D), `name`, and `extra` (`n_delta`, `n`).

Examples

```
set.seed(1)
x <- cumsum(rnorm(1000))
res <- pfd(x)
res$value
```

ph_assumption_test *Proportional-hazards assumption test (Schoenfeld-style)*

Description

Correlates each covariate with rank-transformed event times.

Usage

```
ph_assumption_test(
  survival_times,
  event_indicator,
  covariates,
  covariate_names = NULL
)
```

Arguments

survival_times	Event/censoring times.
event_indicator	1 = event, 0 = censored.
covariates	Covariate matrix.
covariate_names	Optional names.

Value

A list of `morie_specification_test` objects, one per covariate.

phi_coefficient *Phi coefficient for a 2x2 contingency table*

Description

Phi coefficient for a 2x2 contingency table

Usage

```
phi_coefficient(contingency_table)
```

Arguments

contingency_table	2x2 numeric matrix.

Value

A `morie_effect_size`.

`point_biserial_correlation` *Point-biserial correlation*

Description

Point-biserial correlation

Usage

```
point_biserial_correlation(binary, continuous, confidence = 0.95)
```

Arguments

`binary`	0/1 vector.
`continuous`	Numeric vector.
`confidence`	Confidence level.

`prediction_interval` *Prediction interval for a new study (random-effects meta)*

Description

Prediction interval for a new study (random-effects meta)

Usage

```
prediction_interval(estimates, standard_errors, confidence = 0.95)
```

Arguments

`estimates`	Numeric vector of effect-size estimates.
`standard_errors`	Numeric vector of SEs.
`confidence`	Confidence level. Default 0.95.

Value

Numeric c(lower, upper).

`print.morie_audit_result` *Print method for audit results.*

Description

Print method for audit results.

Usage

```
## S3 method for class 'morie_audit_result'
print(x, ...)
```

Arguments

`x`	A `morie_audit_result`.
`...`	Unused.

`print.morie_tps_result` *Pretty-print method for* `morie_tps_result` *objects.*

Description

Pretty-print method for `morie_tps_result` objects.

Usage

```
## S3 method for class 'morie_tps_result'
print(x, ...)
```

Arguments

`x`	A `morie_tps_result` list.
`...`	Ignored.

`print.morie_variable_taxonomy`
Print method for taxonomy entries.

Description

Print method for taxonomy entries.

Usage

```
## S3 method for class 'morie_variable_taxonomy'
print(x, ...)
```

Arguments

`x`	A `morie_variable_taxonomy` object.
`...`	Unused.

`probabilistic_bias_analysis`
Probabilistic (Monte Carlo) sensitivity analysis

Description

Draws bias parameters from prior distributions and returns the distribution of bias-adjusted estimates. Cross-references **episensr** (`episensr::probsens`) for the canonical multi-bias version with separate selection-bias and misclassification-bias models; use **episensr** directly when you need those.

Usage

```
probabilistic_bias_analysis(
  estimate,
  se,
  n_simulations = 10000L,
  bias_parms = NULL,
  seed = 42L
)
```

Arguments

`estimate`	Observed estimate.
`se`	Standard error.
`n_simulations`	Number of MC draws. Default 10000.
`bias_parms`	Named list with `(mean, sd)` pairs for `rr_ud`, `rr_eu`, `prevalence`. Defaults supplied.
`seed`	RNG seed. Default 42.

Value

Named list with bias-adjusted distribution summaries.

`r_effect_size` *Pearson r as an effect size with Fisher-z CI*

Description

Pearson r as an effect size with Fisher-z CI

Usage

```
r_effect_size(x, y, confidence = 0.95)
```

Arguments

`x, y`	Numeric vectors (NA dropped).
`confidence`	Confidence level for CI. Default 0.95.

Value

A `morie_effect_size`.

`r_squared` *Coefficient of determination R^2*

Description

Coefficient of determination R^2

Usage

```
r_squared(x, y)
```

Arguments

`x, y`	Numeric vectors (NA dropped).

Value

A `morie_effect_size`.

r_to_d *Convert Pearson r to Cohen's d*

Description

Convert Pearson r to Cohen's d

Usage

```
r_to_d(r)
```

Arguments

r	Pearson r.

Value

Numeric d.

r_to_or *Convert Pearson r to OR via d*

Description

Convert Pearson r to OR via d

Usage

```
r_to_or(r)
```

Arguments

r	Pearson r.

Value

Numeric OR.

ramsey_reset_test *Ramsey RESET test for functional-form misspecification*

Description

Ramsey RESET test for functional-form misspecification

Usage

```
ramsey_reset_test(y, X, powers = c(2, 3))
```

Arguments

y	Response vector.
X	Design matrix (with intercept).
powers	Integer vector of powers of fitted values to add to the auxiliary regression (default c(2, 3)).

Value

A morie_specification_test.

random_effects_meta *Random-effects (DerSimonian-Laird) meta-analytic pooling*

Description

Random-effects (DerSimonian-Laird) meta-analytic pooling

Usage

```
random_effects_meta(
  estimates,
  standard_errors,
  confidence = 0.95,
  method = "DL"
)
```

Arguments

estimates	Numeric vector of effect-size estimates.
standard_errors	Numeric vector of SEs.
confidence	Confidence level. Default 0.95.
method	Tau^2 estimator. Only "DL" implemented.

Value

A `morie_effect_size` with tau^2, I^2, Q, prediction interval in `extra`.

---

rank_biserial_correlation *Rank-biserial correlation (matched rank version)*

Description

Rank-biserial correlation (matched rank version)

Usage

```
rank_biserial_correlation(x, y, confidence = 0.95)
```

Arguments

`x`, `y`	Numeric vectors (NA dropped).
`confidence`	Confidence level for CI. Default 0.95.

Value

A `morie_effect_size`.

---

rate_ratio *Incidence rate ratio (IRR)*

Description

Incidence rate ratio (IRR)

Usage

```
rate_ratio(events1, person_time1, events2, person_time2, confidence = 0.95)
```

Arguments

`events1`, `person_time1`	Events and person-time in group 1.
`events2`, `person_time2`	Events and person-time in group 2.
`confidence`	Confidence level. Default 0.95.

Value

A `morie_effect_size`.

register_stat_command

Register a stat_command in the package-level registry

Description

Register a stat_command in the package-level registry

Usage

```
register_stat_command(cmd)
```

Arguments

cmd	A `morie_stat_command` constructed by `stat_command`.

Value

The command name, invisibly.

regression_table

Side-by-side regression table for multiple model fits

Description

Side-by-side regression table for multiple model fits

Usage

```
regression_table(
  models,
  exponentiate = FALSE,
  show_ci = TRUE,
  show_stars = TRUE,
  confidence = 0.95,
  digits = 3L,
  model_stats = c("nobs", "rsquared", "aic", "bic", "llf"),
  apa = FALSE,
  output_format = "dataframe",
  title = "Regression Results"
)
```

Arguments

`models`	Named list of fitted models (e.g. `lm`, `glm`).
`exponentiate`	Exponentiate coefficients (for OR / HR).
`show_ci`	Include CI line under each coefficient.
`show_stars`	Append significance stars.
`confidence`	Confidence level for CIs.
`digits`	Decimal places.
`model_stats`	Vector of model-stat keys from `c("nobs","rsquared","aic","bic","llf")`.
`apa`	APA p-value formatting.
`output_format`	Output target.
`title`	Title.

`repeated_cv` *Repeated K-fold cross-validation*

Description

Repeats `.boot_cross_validate()` `n_repeats` times with different RNG seeds and pools the per-fold scores. `caret::trainControl(method = "repeatedcv")` and `rsample::vfold_cv` both implement the same partitioning (cross-referenced).

Usage

```
repeated_cv(
  X,
  y,
  model_fn,
  score_fn,
  n_folds = 10L,
  n_repeats = 10L,
  seed = 42L
)
```

Arguments

`X`	Numeric matrix or data.frame of predictors.
`y`	Numeric or factor outcome vector aligned with rows of `X`.
`model_fn`	Function `(X, y) -> fitted-model` used on each training fold.
`score_fn`	Function `(y_true, y_pred) -> numeric` returning a single performance metric.
`n_folds`	Integer; number of folds per repeat (default 10).
`n_repeats`	Number of repetitions.
`seed`	Integer RNG seed for reproducibility.

Value

A `morie_cv_result` pooling scores across repeats.

See Also

`caret::trainControl`, `rsample::vfold_cv`.

---

repeated_measures_anova *One-way repeated-measures ANOVA*

Description

Sphericity assumption is left to the user; this routine computes the uncorrected F. Pair with `ez::ezANOVA` for GG/HF correction if needed.

Usage

```
repeated_measures_anova(data, outcome, subject, within)
```

Arguments

`data`	Long-format data frame.
`outcome`, `subject`, `within`	Column names.

---

resolve_stat_command *Resolve a command by canonical name or alias*

Description

Resolve a command by canonical name or alias

Usage

```
resolve_stat_command(name)
```

Arguments

`name`	Character scalar.

Value

A `morie_stat_command` or `NULL`.

risk_difference *Risk difference for a 2x2 table*

Description

Risk difference for a 2x2 table

Usage

```
risk_difference(a, b, c, d, confidence = 0.95)
```

Arguments

a, b, c, d	Cell counts.
confidence	Confidence level. Default 0.95.

Value

A morie_effect_size.

risk_ratio *Risk ratio (relative risk) for a 2x2 table*

Description

Risk ratio (relative risk) for a 2x2 table

Usage

```
risk_ratio(a, b, c, d, confidence = 0.95)
```

Arguments

a, b, c, d	Cell counts.
confidence	Confidence level. Default 0.95.

Value

A morie_effect_size.

rosenbaum_bounds *Rosenbaum sensitivity analysis for matched-pair designs*

Description

Phase 1.g delegates to **rbounds** when installed and the `wilcoxon` or `sign` method is requested; otherwise falls back to the base-R normal-approximation implementation originally shipped with rmorie. The `mcnemar` path is always served by the inline binomial formula (rbounds does not expose a McNemar entry point on CRAN).

Usage

```
rosenbaum_bounds(
  treated_outcomes,
  control_outcomes,
  gamma_range = NULL,
  method = "wilcoxon"
)
```

Arguments

`treated_outcomes`	Vector of outcomes for treated units.
`control_outcomes`	Vector of outcomes for matched controls.
`gamma_range`	Numeric vector of Gamma values (default `seq(1, 5, by = 0.25)`).
`method`	One of `"wilcoxon"`, `"sign"`, `"mcnemar"`.

Value

A `morie_rosenbaum_bounds` named-list.

rrint *RR interval series from R-peak sample indices*

Description

Computes the RR (beat-to-beat) interval series in milliseconds from a vector of R-peak sample indices.

Usage

```
rrint(r_peaks, fs)
```

Arguments

`r_peaks`	Integer vector of R-peak sample indices.
`fs`	Sampling frequency in Hz.

Details

Reference: Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996) "Heart rate variability: standards of measurement, physiological interpretation, and clinical use", *Circulation* 93(5):1043–1065.

Value

List with `value` (mean RR in ms), `name`, and `extra` (`rr_ms`, `mean_rr`, `std_rr`, `n_intervals`).

Examples

```
rr <- rrint(c(100, 350, 600, 850, 1100), fs = 250)
rr$value
```

`run_stat_command` *Run a command's REPL handler with positional/keyword arguments*

Description

Run a command's REPL handler with positional/keyword arguments

Usage

```
run_stat_command(name, ...)
```

Arguments

`name`	Command name or alias.
`...`	Arguments forwarded to the REPL handler.

Value

Whatever the handler returns. Stops with an informative error if the command is not registered.

runs_test — *Wald-Wolfowitz runs test for randomness*

Description

Wald-Wolfowitz runs test for randomness

Usage

```
runs_test(x, cutoff = NULL)
```

Arguments

x	Numeric sequence.
cutoff	Cut-off (median by default).

sampen — *Sample entropy*

Description

Sample entropy (SampEn) of a 1-D signal: $-\log(A/B)$, where A counts template-vector matches at embedding dimension $m+1$ and B at dimension m, with Chebyshev distance tolerance $r \cdot \mathrm{sd}(x)$.

Usage

```
sampen(x, m = 2L, r = 0.2)
```

Arguments

x	Numeric vector.
m	Embedding dimension (default 2).
r	Tolerance as fraction of sd (default 0.2).

Details

Reference: Richman, J.S. & Moorman, J.R. (2000) "Physiological time- series analysis using approximate entropy and sample entropy", *Am. J. Physiol. Heart Circ. Physiol.* 278(6):H2039–H2049 (refines Pincus, S.M. (1991), *Proc. Natl. Acad. Sci. USA* 88:2297).

Value

List with `value` (SampEn), `name`, and `extra` (`m`, `r`, `tolerance`, `A`, `B`).

Examples

```
set.seed(1)
x <- sin(seq(0, 10 * pi, length.out = 500)) + 0.1 * rnorm(500)
res <- sampen(x)
res$value
```

---

score_data_quality *Multi-dimensional data quality scores*

Description

Multi-dimensional data quality scores

Usage

```
score_data_quality(
  data,
  date_cols = NULL,
  freshness_days = 365L,
  key_cols = NULL,
  consistency_rules = NULL
)
```

Arguments

`data`	Data frame.
`date_cols`	Datetime column names for timeliness.
`freshness_days`	Days for full timeliness score.
`key_cols`	Columns that should be unique together.
`consistency_rules`	List of functions `(df) -> logical(1)`.

---

score_test *Score (Lagrange multiplier) test*

Description

Score (Lagrange multiplier) test

Usage

```
score_test(score_vector, information_matrix)
```

Arguments

`score_vector`	Score vector evaluated under H0.
`information_matrix`	Information matrix under H0.

Value

A `morie_specification_test`.

`semi_partial_correlation` *Semi-partial (part) correlation*

Description

Semi-partial (part) correlation

Usage

```
semi_partial_correlation(x, y, covariates)
```

Arguments

`x`, `y`	Numeric vectors of interest.
`covariates`	Matrix or data frame of covariates.

`SemiparKernels` *Object-style wrapper for the semiparametric kernel toolkit*

Description

Returns a list of closures bound to the same backend (always pure R in this port; the Python module additionally supports a C backend).

Usage

```
SemiparKernels()
```

Value

A list with class `morie_semipar_kernels` carrying methods `nw_regression`, `local_linear`, `kde`, `silverman_bandwidth`, `loocv_bandwidth`, `kernel_cond_moments`, plus a `backend` string.

`sensitivity` *Sensitivity analysis for causal inference assumptions*

Description

Tools to assess the robustness of causal effect estimates to unmeasured confounding, model specification, and other threats to internal validity. Includes Rosenbaum bounds, the E-value family, Ding-VanderWeele bias formulas, tipping-point analysis, omitted- variable bias (Cinelli-Hazlett), Manski bounds, probabilistic (Monte-Carlo) bias analysis, and specification curve analysis.

Details

Wraps CRAN **EValue**, **tipr**, **sensemakr**, **specr**, **rbounds**, **episensr**, and **konfound** when available; falls back to base-R closed-form implementations otherwise.

References

Rosenbaum (2002); VanderWeele & Ding (2017); Cinelli & Hazlett (2020); Manski (1990); Ding & VanderWeele (2016).

`sensitivity_rosenbaum`

Rosenbaum bounds sensitivity analysis (data-frame interface)

Description

Thin wrapper over `rbounds::psens()` when **rbounds** is installed (rank-matched-pair signed-rank bounds across a Gamma grid). Without **rbounds**, falls back to a base R normal-approximation Wilcoxon signed-rank computation.

Usage

```
sensitivity_rosenbaum(
  data,
  treatment,
  outcome,
  covariates,
  gamma_range = c(1, 3),
  n_gamma = 20L
)
```

Arguments

`data`	Data frame with treatment + outcome columns.
`treatment`	Binary treatment column (0/1).
`outcome`	Outcome column.
`covariates`	Covariates (used only for matching approximation, here a simple rank-match).
`gamma_range`	c(min, max) of Gamma. Default c(1, 3).
`n_gamma`	Number of Gamma values. Default 20.

Value

Data frame with `Gamma`, `p_lower`, `p_upper`.

`sensitivity_summary` *Generate a comprehensive sensitivity-analysis summary*

Description

Produces a tidy data.frame with the estimate, CI, p-value, applicable E-values (RR / OR / HR), and a tipping-point delta.

Usage

```
sensitivity_summary(
  estimate,
  se,
  rr = NULL,
  odds_ratio = NULL,
  hazard_ratio = NULL,
  prevalence = NULL
)
```

Arguments

`estimate`	Treatment-effect estimate.
`se`	Standard error.
`rr`, `odds_ratio`, `hazard_ratio`	Optional effect on each scale.
`prevalence`	Outcome prevalence (for OR-to-RR).

Value

A data.frame with `metric, value`.

`sgolay` *Savitzky-Golay smoothing (direct alias)*

Description

Direct short-name export of the Savitzky-Golay smoother (matches the Python `morie.signal.sgolay` name). For the long-form, see `morie_sgolay_smooth()`, which this function delegates to.

Usage

```
sgolay(x, window = 11L, polyorder = 3L)
```

Arguments

`x`	Numeric vector.
`window`	Window length (odd, default 11).
`polyorder`	Polynomial order (default 3).

Details

Reference: Savitzky, A. & Golay, M.J.E. (1964) "Smoothing and differentiation of data by simplified least-squares procedures", *Anal. Chem.* 36(8):1627–1639.

Value

List with `filtered`, `name`, `fs`, `n_samples`, `extra` (`window`, `polyorder`).

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  set.seed(1)
  x <- sin(seq(0, 2 * pi, length.out = 200)) + 0.1 * rnorm(200)
  res <- sgolay(x)
  length(res$filtered)
}
```

sidak — *Sidak FWER correction*

Description

Slightly less conservative than Bonferroni under independence. Computes $1 - (1 - p)^m$ directly (closed form); the **mutoss** package offers an equivalent step-down variant via `mutoss::SidakSD`.

Usage

```
sidak(p_values, alpha = 0.05, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`labels`	Optional character vector of test labels.

silverman_bandwidth — *Silverman rule-of-thumb bandwidth*

Description

Returns h equal to 0.9 times min of sigma-hat and IQR over 1.34 times n to the negative one-fifth.

Usage

```
silverman_bandwidth(x)
```

Arguments

`x`	Numeric data vector.

Value

Bandwidth (numeric scalar).

References

Silverman, B. W. (1986), p. 48.

simes_combined *Simes test for the global null*

Description

Simes test for the global null

Usage

```
simes_combined(p_values)
```

Arguments

p_values	Numeric vector of raw p-values.

spearman_correlation *Spearman rank correlation*

Description

Spearman rank correlation

Usage

```
spearman_correlation(x, y, confidence = 0.95)
```

Arguments

x, y	Numeric vectors.
confidence	Confidence level.

specification_curve *Specification curve analysis*

Description

Estimates the treatment effect across many reasonable model specifications to assess robustness. Combines covariate sets x sample filters x model families. Cross-references **specr** (`specr::specr`) as the canonical modern implementation with built-in plotting; use **specr** directly when you want the published specification-curve plot.

Usage

```
specification_curve(
  data,
  outcome,
  treatment,
  covariate_sets,
  sample_filters = NULL,
  model_types = NULL,
  alpha = 0.05
)
```

Arguments

`data`	Analysis data.frame.
`outcome`	Outcome variable name.
`treatment`	Treatment variable name.
`covariate_sets`	List of character vectors (one per spec).
`sample_filters`	Optional. Accepted shapes (for Python<->R parity): (a) `list(list(name = "...", fn = function(df) ...), ...)` (R native), (b) `list(c("name", fn), ...)` or `list(list("name", fn), ...)` (Python `list[tuple[str, callable]]` shape — positional pair). Default: full sample only.
`model_types`	Character vector of model families: `"ols"`, `"logistic"`, `"robust"`. Default `c("ols")`.
`alpha`	Significance level. Default 0.05.

Value

A `morie_spec_curve` named-list.

`standardized_coefficients` *Standardised regression coefficients (beta weights)*

Description

Standardises X and y to zero mean and unit variance before OLS via `stats::lm`.

Usage

```
standardized_coefficients(X, y)
```

Arguments

`X`	Predictor matrix or data.frame (n x p).
`y`	Outcome vector.

Value

A data.frame with columns `variable, beta, se, t, p_value`.

`stat_bridge_exec` *Execute a single command and return the resulting text*

Description

Execute a single command and return the resulting text

Usage

```
stat_bridge_exec(cmd_str)
```

Arguments

`cmd_str`	A whitespace-delimited command line, e.g. `"bonferroni 0.01 0.04 0.05"`.

Value

Captured handler output as a single string.

`stat_bridge_fn_info` *Inspect a single command by name*

Description

Inspect a single command by name

Usage

```
stat_bridge_fn_info(name)
```

Arguments

`name`	Command name or alias.

Value

Multi-line description string or an explanatory error string.

stat_bridge_fn_search

Search the registry for matching commands

Description

Search the registry for matching commands

Usage

```
stat_bridge_fn_search(query, max_results = 20L)
```

Arguments

`query`	Free-text query; matched against names, categories, descriptions, and aliases.
`max_results`	Cap on the number of matches returned.

Value

Multi-line summary string.

stat_bridge_help

Formatted text dump of the command registry

Description

Formatted text dump of the command registry

Usage

```
stat_bridge_help()
```

Value

A length-1 character string.

stat_bridge_main *Command-line dispatcher*

Description

Mirrors `python -m morie.stat_bridge <mode> [...]` so the same invocation pattern is available via `Rscript -e`.

Usage

```
stat_bridge_main(args = NULL)
```

Arguments

`args`	Character vector of CLI arguments (mode + parameters). When `NULL`, defaults to `commandArgs(trailingOnly = TRUE)`.

Details

Recognised modes: `"registry-json"`, `"help"`, `"exec"`, `"fn-info"`, `"fn-search"`, `"verify"`.

Value

Invisibly returns the printed text; primarily called for side effects (printing to stdout).

stat_bridge_registry_json *JSON enumeration of all registered commands*

Description

JSON enumeration of all registered commands

Usage

```
stat_bridge_registry_json()
```

Value

A length-1 character vector containing JSON text.

stat_bridge_verify *Self-test enumeration helper*

Description

Calls every registered handler with no arguments inside `tryCatch`, reporting which entries can be invoked safely. Intended to be called from CI smoke tests.

Usage

```
stat_bridge_verify()
```

Value

A data.frame with columns `name`, `ok`, `message`.

stat_command *Construct a stat_command entry*

Description

Construct a stat_command entry

Usage

```
stat_command(
  name,
  category,
  usage,
  description,
  handler_repl,
  handler_stat = NULL,
  aliases = character(0),
  module = "",
  is_compound = FALSE,
  is_r_bridge = FALSE
)
```

Arguments

`name`	Canonical command name (character scalar).
`category`	Category label used for grouping.
`usage`	One-line usage string for help screens.
`description`	Short description.
`handler_repl`	Function implementing the command.

`handler_stat`	Optional terminal handler taking `(parts, log, store)`. Defaults to a wrapper around `handler_repl`.
`aliases`	Character vector of additional names.
`module`	Source module string (informational).
`is_compound`	Logical; flags compound workflows.
`is_r_bridge`	Logical; flags Python <-> R bridge calls.

Value

A list with class `morie_stat_command`.

`statistics` *Comprehensive hypothesis testing suite for epidemiological research*

Description

R port of the Python module `morie.statistics`. Every function returns a named `list` (class `"morie_test_result"`) carrying the test statistic, p-value, degrees of freedom, confidence interval, effect size, point estimate, sample size and a free-form `extra` list, so downstream code can post-process programmatically.

Details

Categories:

- Location: `one_sample_ttest`, `two_sample_ttest`, `welch_ttest`, `paired_ttest`
- ANOVA / non-parametric ANOVA: `one_way_anova`, `two_way_anova`, `repeated_measures_anova`, `friedman_test` (`stats::kruskal.test` for K-W)
- Chi-squared family: `chi2_goodness_of_fit`, `chi2_independence`, `mcnemar_test`, `cochrans_q`
- Correlation: `pearson_correlation`, `spearman_correlation`, `kendall_correlation`, `point_biserial_correlation`, `partial_correlation`, `semi_partial_correlation`
- Non-parametric: `mann_whitney_u`, `wilcoxon_signed_rank`, `ks_test_one_sample`, `ks_test_two_sample`, `levene_test`, `bartlett_test`, `runs_test` (`nortest::ad.test` for Anderson-Darling)
- Normality: `dagostino_pearson`, `lilliefors_test` (`stats::shapiro.test` for Shapiro-Wilk, `tseries::jarque.bera.test` for Jarque-Bera)
- Proportions: `one_proportion_ztest`, `two_proportion_ztest`, `fisher_exact_test`
- Agreement: `cohens_kappa`, `intraclass_correlation` (`irr::kappam.fleiss` for Fleiss' kappa)
- Convenience: `normality_suite`, `variance_equality_suite`, `correlation_matrix`, `auto_test`

storey_q — *Storey q-value procedure (adaptive FDR)*

Description

Estimates the proportion of true null hypotheses (pi0) and tightens the BH thresholds by that factor. Delegates to `qvalue::qvalue` (Bioconductor) when installed; otherwise falls back to an inline Storey-style cutoff so the wrapper keeps working on CRAN-only installs.

Usage

```
storey_q(p_values, alpha = 0.05, lambda_param = 0.5, labels = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`alpha`	Significance level.
`lambda_param`	Tuning parameter in (0, 1) for the pi0 estimator.
`labels`	Optional character vector of test labels.

stouffer_combined — *Stouffer's z-score method*

Description

Delegates to `poolr::stouffer` when installed and no weights are supplied; otherwise computes the weighted z-sum inline.

Usage

```
stouffer_combined(p_values, weights = NULL)
```

Arguments

`p_values`	Numeric vector of raw p-values.
`weights`	Optional non-negative weights (any scale).

`subsampling` *Subsampling inference (Politis, Romano & Wolf)*

Description

Draws without replacement at a smaller sample size; valid under weaker conditions than the bootstrap. No clean CRAN function exposes the same `(data, statistic, subsample_size, n_subsamples)` API; **np**'s `npsubsample` is closest but is kernel-specific. Kept as an in-house implementation and flagged novel/no-clean-CRAN-equivalent.

Usage

```
subsampling(
  data,
  statistic,
  subsample_size = NULL,
  n_subsamples = 1000L,
  ci_level = 0.95,
  seed = 42L
)
```

Arguments

`data`	Numeric vector or matrix.
`statistic`	Function returning a scalar.
`subsample_size`	Subsample size; default `floor(n^0.7)`.
`n_subsamples`	Number of subsamples.
`ci_level`	Confidence level.
`seed`	Random seed.

Value

A `morie_bootstrap_result`.

`substance_categories` *Substance Categories*

Description

Canonical substance category mapping used across CSUS HealthInfobase data files. Maps short keys to human-readable labels and source filenames.

Usage

```
substance_categories
```

Format

A data.frame with columns:

key Short key (e.g., `"alcohol"`, `"cannabis"`)

label Display label (e.g., `"Alcohol"`, `"Cannabis"`)

source_file Filename in healthinfobase/CSUS/ directory

Source

Canadian Substance Use Survey (CSUS) via Health Infobase Canada.

Examples

```
data(substance_categories)
substance_categories$label
```

---

summary_statistics_table

Descriptive statistics for a set of variables

Description

Descriptive statistics for a set of variables

Usage

```
summary_statistics_table(
  data,
  variables = NULL,
  stats = c("n", "mean", "sd", "median", "min", "max", "missing"),
  digits = 2L,
  output_format = "dataframe",
  title = "Summary Statistics"
)
```

Arguments

`data`	Data frame.
`variables`	Variable names (auto-detect numeric if NULL).
`stats`	Vector of statistic names.
`digits`	Decimal places.
`output_format`	Output target.
`title`	Title.

`table1` *Table 1 (baseline characteristics) stratified by group*

Description

Table 1 (baseline characteristics) stratified by group

Usage

```
table1(
  data,
  group_col = NULL,
  continuous_vars = NULL,
  categorical_vars = NULL,
  continuous_summary = c("mean_sd", "median_iqr", "mean_ci"),
  show_p = TRUE,
  show_smd = TRUE,
  show_missing = TRUE,
  weights = NULL,
  digits = 2L,
  apa = FALSE,
  output_format = "dataframe",
  title = "Table 1. Baseline Characteristics"
)
```

Arguments

`data`	Data frame.
`group_col`	Column defining groups, or NULL.
`continuous_vars`	Continuous variable names (auto-detect numeric non-group columns if NULL).
`categorical_vars`	Categorical names (auto-detect character / factor / logical if NULL).
`continuous_summary`	"mean_sd", "median_iqr" or "mean_ci".
`show_p`	Include p-value column.
`show_smd`	Include SMD column (2 groups only).
`show_missing`	Include missing count.
`weights`	Column name for survey weights or NULL.
`digits`	Decimal places.
`apa`	APA-style p-value formatting.
`output_format`	"dataframe", "latex", "html", "markdown", "text", "csv".
`title`	Table title.

tables_pub *Publication-ready table generation*

Description

R port of the Python module `morie.tables_pub`. Builds Table 1 (baseline characteristics), regression tables, odds-ratio and hazard-ratio tables, correlation matrices, model comparison tables, ANOVA tables, summary-statistics tables and treatment-effect tables.

Details

Output rendering goes through **knitr::kable** for "latex", "html", "markdown", "pipe" and "rst" formats; "dataframe" returns the raw `data.frame`, and "text" returns `utils::capture.output` on the frame. The **gt** package is supported as an optional richer-output backend when installed (Suggests-gated).

Functions consume R-native model objects:

- `regression_table` accepts `lm`, `glm` or any object responding to `coef`, `vcov` and `confint`.
- `odds_ratio_table` accepts a fitted `glm` with `family = binomial()`.
- `hazard_ratio_table` accepts a `coxph` fit, or per-parameter `beta`, `se` and `p` vectors.
- `anova_table` wraps `stats::anova` / `car::Anova`.

temporal_validate *Train on earlier data, test on later data*

Description

Train on earlier data, test on later data

Usage

```
temporal_validate(
  fit_fn,
  predict_fn,
  X,
  y,
  date_col,
  split_date = NULL,
  split_quantile = 0.7,
  scoring = "roc_auc"
)
```

Arguments

`fit_fn`, `predict_fn`	As in `cross_validate`.
`X`	Data frame including `date_col`.
`y`	Target vector.
`date_col`	Name of date column in `X`.
`split_date`	Date to split on, or NULL.
`split_quantile`	Quantile of dates (if `split_date` is NULL).
`scoring`	Scoring metric.

`tippett_combined` *Tippett's minimum-p method*

Description

Delegates to `poolr::tippett` when installed; otherwise computes the closed form inline.

Usage

```
tippett_combined(p_values)
```

Arguments

`p_values`	Numeric vector of raw p-values.

`tipping_point_analysis` *Tipping-point analysis for missing-data sensitivity*

Description

How much would unobserved outcomes need to differ from observed ones for the treatment effect to become non-significant? Phase 1.g cross-references **tipr** for the unmeasured-confounder family of tipping-point calculations (see also `morie_sensitivity_tipping_point`).

Usage

```
tipping_point_analysis(
  estimate,
  se,
  n_treated,
  n_control,
  delta_range = NULL,
  outcome_type = "continuous"
)
```

Arguments

estimate	Observed treatment effect.
se	Standard error of the estimate.
n_treated	Number of treated units.
n_control	Number of control units.
delta_range	Numeric vector of bias parameters (default seq(-3|est|, 3|est|, length.out = 101)).
outcome_type	"continuous" or "binary" (advisory only).

Value

A morie_tipping_point named-list.

---

tps_crime *Cross-category crime analyses for Toronto Police Service (TPS) datasets*

Description

R-side port of morie.tps_crime. Where the Python module uses TPS_REGISTRY + load_tps_dataset to materialise per-category data.frames, the R-side callables here accept a named list of pre-loaded data.frames (one entry per TPS category) via the dfs argument. Callers are responsible for loading the CSVs (e.g. via utils::read.csv or readr::read_csv) and passing them in keyed by canonical TPS category name (*e.g.* "Assault", "Homicides", "BicycleTheft").

Details

Callables:

- morie_tps_yoy_panel(): side-by-side year-over-year panel across TPS categories.
- morie_tps_composite_index(): per-neighbourhood composite crime-risk index (sum of z-standardised counts, optionally weighted).
- morie_tps_bivariate_morans_i(): bivariate Moran's I between two TPS categories on a shared HOOD_158 footprint using a k-NN row-standardised spatial weights matrix.
- morie_tps_category_correlation_matrix(): Pearson r on per-hood incident counts across all supplied categories.

Each callable returns a named list with class c("morie_tps_result", "morie_rich_result", "list") carrying title, summary_lines, tables (when applicable), interpretation, warnings, and a free-form payload.

`tps_csi` *Statistics Canada Crime Severity Index (CSI) weights for TPS data*

Description

R-side port of `morie.tps_csi`. The Crime Severity Index (Wallace et al., 2009; Statistics Canada Catalogue 85-004-X) weights each *Criminal Code* offence by the product of the average sentence length (days) and the proportion of offenders incarcerated, so that violent offences with high incarceration rates and long sentences contribute disproportionately to a city's per-capita CSI score.

Details

This file exposes the weights used for the 9 Toronto Police Service open-data categories (Assault, Auto Theft, Bicycle Theft, Break and Enter, Homicide, Robbery, Shooting and Firearm Discharges, Theft from Motor Vehicle, Theft Over) and provides per-year + per-neighbourhood CSI aggregates.

Important caveats:

1. TPS open-data categories aggregate over multiple Criminal Code sub-offences. The weights here are representative blends reflecting the typical distribution of sub-offences within each TPS category for FY2023; for an exact reproduction of Statistics Canada's CSI for the City of Toronto one must work directly from the CCJS UCR microdata, which is not in TPS open data.
2. Weights are pinned to the last published StatsCan methodology update (*Reweighting the Crime Severity Index*, Catalogue 85-004-X) and the Toronto-specific override tables in the CCJS Annual Statistics 2023. Newer revisions (StatsCan revises every 5 years) may shift values by 5-15\ ordering. Override via the `weights` argument.
3. Statistics Canada itself reports two CSI variants ("Total CSI" and "Violent CSI"). Functions here default to Total but accept `variant = "violent"` to use violent-only weights, where non-violent categories (B&E, theft) are zeroed.

References:

Wallace, M., Turner, J., Babyak, C., & Matarazzo, A. (2009). Measuring Crime in Canada: Introducing the Crime Severity Index and Improvements to the Uniform Crime Reporting Survey. Statistics Canada Catalogue 85-004-X.

Statistics Canada (2024). Crime Severity Index, Census Metropolitan Areas, 2023. Catalogue 35-10-0190-01.

tps_spatial *Spatial analyses for TPS crime data*

Description

R parity of `morie.tps_spatial`: Moran's I (global), LISA (local Moran's Ii) for hot/cold spots, and 2-D kernel density estimation of incident lat/long. Each function accepts a `data.frame` of incident-level rows with a neighbourhood id column plus WGS84 lat/long columns, and returns a named `list` carrying numeric outputs alongside a multi-paragraph `interpretation` so the result prints in a notebook without further post-processing.

Details

Spatial weights are built with an internal base-R k-nearest- neighbours routine; if the optional **FNN** package is installed it is used for the KNN graph. The 2-D kernel density estimator prefers **MASS**`::kde2d` when available, otherwise falls back to a Gaussian density evaluated at the observation points. If **spdep** is installed, callers can delegate the global Moran's I test to `spdep::moran.test` via the `use_spdep = TRUE` switch.

Functions:

- `morie_tps_morans_i_neighbourhood`: global Moran's I on neighbourhood-level incident counts.
- `morie_tps_local_morans_i`: LISA (local Moran's Ii) per neighbourhood with HH/LL/HL/LH quadrant classification.
- `morie_tps_kde_density`: 2-D kernel density estimate of geocoded incident points.

tps_spatial_advanced *Heavyweight spatial statistics for TPS data*

Description

R parity of `morie.tps_spatial_advanced`. Builds on `tps_spatial` (global Moran's I, LISA, KDE) with:

Details

- `morie_tps_ripley_k`: Ripley's K function for point-pattern clustering at multiple radii.
- `morie_tps_getis_ord_g_star`: local Getis-Ord Gi* hot/cold-spot z-scores.
- `morie_tps_dbscan_clusters`: density-based clusters on lat/long (via **dbscan**, optional).
- `morie_tps_polygon_morans_i`: polygon-aware Moran's I from an **sf** object's actual polygon centroids (instead of the centroid-only k-NN approximation in `morie_tps_morans_i_neighbourhood`)

- `morie_tps_bivariate_moran`: bivariate Moran's I between two attributes at the same polygons.
- `morie_tps_moran_sweep_heatmap`: a (category x year) sweep of polygon Moran's I.

Polygon functions accept either an **sf** object (gated with `requireNamespace("sf")`) or a plain data.frame carrying precomputed centroid columns. KNN graphs prefer **FNN**; DBSCAN requires the optional **dbscan** package; spatial autocorrelation tests can optionally be delegated to **spdep**.

`treatment_effect_table` *Summary table of causal effect estimates from multiple estimators*

Description

Summary table of causal effect estimates from multiple estimators

Usage

```
treatment_effect_table(
  estimators,
  digits = 3L,
  output_format = "dataframe",
  title = "Treatment Effect Estimates"
)
```

Arguments

`estimators`	Named list of lists; each inner list provides numeric `estimate`, `se`, `ci_lower`, `ci_upper` and `p_value`.
`digits`	Decimal places.
`output_format`	Output target.
`title`	Title.

`two_proportion_ztest` *Two-sample z-test for the difference in proportions*

Description

Two-sample z-test for the difference in proportions

Usage

```
two_proportion_ztest(count1, nobs1, count2, nobs2, confidence = 0.95)
```

Arguments

`count1`, `nobs1`	First sample.
`count2`, `nobs2`	Second sample.
`confidence`	Confidence level.

`two_sample_ttest` *Independent two-sample t-test (equal or unequal variance)*

Description

Independent two-sample t-test (equal or unequal variance)

Usage

```
two_sample_ttest(x, y, equal_var = TRUE, confidence = 0.95)
```

Arguments

`x`, `y`	Numeric vectors.
`equal_var`	If FALSE, use Welch's correction.
`confidence`	Confidence level.

`two_way_anova` *Two-way factorial ANOVA (Type-II SS)*

Description

Uses base R `aov`, then `drop1` for Type-II sums of squares.

Usage

```
two_way_anova(data, outcome, factor_a, factor_b)
```

Arguments

`data`	A data frame.
`outcome`	Name of dependent-variable column.
`factor_a`, `factor_b`	Names of factor columns.

validate_schema *Validate a data frame against a list of column rules*

Description

Validate a data frame against a list of column rules

Usage

```
validate_schema(data, rules, raise_on_error = FALSE)
```

Arguments

data	A data frame.
rules	List of column_rule objects.
raise_on_error	If TRUE, throw on first error.

validation *Data and model validation framework*

Description

R port of the Python module morie.validation: schema validation, data quality scoring, cross-validation, calibration / discrimination / decision-curve analysis, overfitting detection, temporal / external validation, and reproducibility manifests.

Details

Most callables return a named list (class "morie_validation_result") so the R caller does not need **S4** or **R6**. Model-fitting routines accept a user-supplied fit_fn of signature function(X, y) -> model and a predict_fn of signature function(model, X) -> prob; this keeps the port framework-agnostic (works with **glm**, **glmnet**, **randomForest**, **xgboost**, etc.).

vancouver_crime_adjacent *Bundled Vancouver Open Data crime-adjacent civic datasets*

Description

Phase 3DDD1. Five small fixtures harvested live from opendata.vancouver.ca for offline reproducibility – chosen to surface neighbourhood-level civic context useful in carceral / policing analysis even though VPD itself publishes crime data separately (see `morie_datasets_vpd_crime()`).

Phase 3EEE3. Bundled 27-row snapshot of City-run community centres. Useful as an "anchor institutions" overlay for analyses of neighbourhood-level crime + social-service access.

Phase 3EEE3. Bundled 91-row snapshot of community + farmers markets across Vancouver. Useful for food-access / quality-of-life overlays.

Phase 3EEE3. Bundled 100-row sample of designated disability parking locations across Vancouver (out of 159 total).

Phase 3EEE3. Bundled 100-row sample of Vancouver's public art registry (out of 747 total) – artist, install year, neighbourhood, primary material. Useful as a CPTED-style "place-making" overlay variable.

Usage

```
morie_datasets_vancouver_graffiti(offline = TRUE, max_features = NULL)

morie_datasets_vancouver_noise_control_areas(
  offline = TRUE,
  max_features = NULL
)

morie_datasets_vancouver_homeless_shelters(offline = TRUE, max_features = NULL)

morie_datasets_vancouver_property_use_inspection_districts(
  offline = TRUE,
  max_features = NULL
)

morie_datasets_vancouver_fire_halls(offline = TRUE, max_features = NULL)

morie_datasets_vancouver_community_centres(offline = TRUE, max_features = NULL)

morie_datasets_vancouver_community_food_markets(
  offline = TRUE,
  max_features = NULL
)

morie_datasets_vancouver_disability_parking(
```

```
  offline = TRUE,
  max_features = NULL
)

morie_datasets_vancouver_public_art(offline = TRUE, max_features = NULL)
```

Arguments

`offline`	If `TRUE` (default), reads the bundled CSV; if `FALSE`, paginates the live catalog endpoint.
`max_features`	Optional row cap.

Details

Loader	Dataset slug
`morie_datasets_vancouver_graffiti()`	`graffiti`
`morie_datasets_vancouver_noise_control_areas()`	`noise-control-areas`
`morie_datasets_vancouver_homeless_shelters()`	`homeless-shelter-locations`
`morie_datasets_vancouver_property_use_inspection_districts()`	`property-use-inspection-districts`
`morie_datasets_vancouver_fire_halls()`	`fire-halls`

All loaders accept the same `offline = TRUE` (default) / `max_features` interface as the other morie dataset wrappers.

`vargha_delaney_a` *Vargha-Delaney A statistic*

Description

Vargha-Delaney A statistic

Usage

```
vargha_delaney_a(x, y, confidence = 0.95)
```

Arguments

`x, y`	Numeric vectors (NA dropped).
`confidence`	Confidence level for CI. Default 0.95.

Value

A `morie_effect_size`.

`variable_taxonomy` *Per-variable taxonomy + dispatcher (R mirror of morie.variable_taxonomy)*

Description

Classifies every column in OTIS / ARSAU datasets by Stevens-1946 level of measurement (nominal / ordinal / interval / ratio + the practical extensions boolean / date / datetime / identifier / free-text), cardinality, functional role (identifier / outcome / covariate / weight / metadata), and cross-year safety.

Details

Drives a method dispatcher (`morie_recommended_summary`, `morie_recommended_pair_test`) that picks the right statistical analysis per variable based on its measurement level.

Hard-coded invariant overrides (the data dictionary itself states these, but we encode them in code so analyses cannot accidentally violate them):

- OTIS `UniqueIndividual_ID`: random per-fiscal-year reassignment -> `cross_year_safe = FALSE`, `role = "identifier"`. Cross-year joins on this column are statistically meaningless.
- ARSAU `BatchFileName` / `Indiv_Index`: per- incident identifiers -> `role = "identifier"`.
- ARSAU `IndivInjuries_PhysicalInjuries`: boolean injury outcome -> `role = "outcome"`.

References

Stevens, S.S. (1946) "On the theory of scales of measurement." *Science*, 103(2684), 677-680.

Velleman, P.F. and Wilkinson, L. (1993) "Nominal, ordinal, interval, and ratio typologies are misleading." *The American Statistician*, 47(1), 65-72.

`variance_equality_suite`

Run a suite of homogeneity-of-variance tests

Description

Run a suite of homogeneity-of-variance tests

Usage

```
variance_equality_suite(...)
```

Arguments

...	Two or more numeric vectors.

`variance_ratio` *Variance ratio (F-test for equality of variances)*

Description

Variance ratio (F-test for equality of variances)

Usage

```
variance_ratio(x, y, confidence = 0.95)
```

Arguments

`x`, `y`	Numeric vectors (NA dropped).
`confidence`	Confidence level for CI. Default 0.95.

Value

A `morie_effect_size`.

`wald_test` *Wald test for linear restrictions on parameters*

Description

Tests H0: `R %*% beta = r`.

Usage

```
wald_test(estimates, vcov, R = NULL, r = NULL)
```

Arguments

`estimates`	Parameter estimates.
`vcov`	Variance-covariance matrix.
`R`	Optional restriction matrix (default identity).
`r`	Optional restriction vector (default zeros).

Value

A `morie_specification_test`.

`welch` *Welch power spectral density*

Description

Welch's averaged periodogram PSD: split into segments, window (Hanning), periodogram each, average. Delegates to `oce::pwelch()` if available, otherwise computes a Hanning-windowed, 50%-overlap implementation in base R.

Usage

```
welch(x, fs, nperseg = 256L)
```

Arguments

`x`	Numeric vector (1-D signal).
`fs`	Sampling frequency in Hz.
`nperseg`	Segment length (default 256).

Details

Reference: Welch, P.D. (1967) "The use of fast Fourier transform for the estimation of power spectra", *IEEE Trans. Audio Electroacoust.* AU-15(2):70–73.

Value

List with `filtered` (PSD), `name`, `fs`, `n_samples`, and `extra` (`freqs`).

Examples

```
set.seed(1)
t <- seq(0, 1, length.out = 1024)
x <- sin(2 * pi * 50 * t) + 0.3 * rnorm(length(t))
res <- welch(x, fs = 1024)
length(res$filtered)
```

welch_ttest — *Welch's t-test (convenience wrapper)*

Description

Welch's t-test (convenience wrapper)

Usage

```
welch_ttest(x, y, confidence = 0.95)
```

Arguments

x, y	Numeric vectors.
confidence	Confidence level.

wilcoxon_signed_rank — *Wilcoxon signed-rank test (one-sample or paired)*

Description

Wilcoxon signed-rank test (one-sample or paired)

Usage

```
wilcoxon_signed_rank(x, y = NULL, alternative = "two.sided")
```

Arguments

x	Numeric vector.
y	Optional paired vector.
alternative	One of "two.sided", "less", "greater".

`wild_bootstrap` *Wild bootstrap for linear regression with heteroskedasticity*

Description

Multiplies the residuals by random weights (Rademacher or Mammen) and refits OLS. `sandwich::vcovBS` implements the standard wild bootstrap variance-covariance and `fwildclusterboot::boottes` adds cluster-wild p-values; both are cross-referenced here. The inline implementation is retained because rmorie's API returns the resampled coefficient distribution (not just a vcov), which is what downstream MRM analyses consume.

Usage

```
wild_bootstrap(
  y,
  X,
  statistic_idx = 2L,
  n_boot = 999L,
  ci_level = 0.95,
  weight_distribution = "rademacher",
  seed = 42L
)
```

Arguments

`y`	Numeric response vector.
`X`	Numeric design matrix (include an intercept column).
`statistic_idx`	Column index of the coefficient of interest (1-based).
`n_boot`	Number of replicates.
`ci_level`	Confidence level.
`weight_distribution`	`"rademacher"` or `"mammen"`.
`seed`	Random seed.

Value

A `morie_bootstrap_result`.

See Also

`sandwich::vcovBS`, `fwildclusterboot::boottest`, `morie_did_wild_cluster_bootstrap()`.

Index

∗ **datasets**
CCRSO_TABLE1_RELEASES, 53
CCRSO_TABLE2_FLOW, 53
CCRSO_TABLE3_AGE, 54
ckan_metadata, 57
dataset_catalog, 70
kernel-codes, 108
MORIE_FAIRNESS_CANONICAL_FIELDS, 481
MORIE_SIUIAP_AFFIDAVITS, 816
MORIE_SIUIAP_CRIMSL_REPORTS, 817
MORIE_SIUIAP_ORIGINAL_PANEL_2019_2020, 817
MORIE_SIUIAP_PANEL_MANDATE, 818
MORIE_SIUIAP_PANEL_MEMBERS, 818
MORIE_SIUIAP_REPORTS, 819
MORIE_SIUIAP_URL, 819
morie_tps_layer_urls, 910
MORIE_TPS_REGISTRY, 923
MORIE_TPS_SUPPORTED_FORMATS, 937
substance_categories, 1079
.morie_db_index_registry(), *370*

adjust_p_values, 34
all_stat_command_names, 35, *852*
analyze_doob_full_affidavit, 35
analyze_doob_table1_releases, 35
analyze_doob_table2_flow, 36
analyze_doob_table3_age_overrepresentation, 36
anova_table, 37
arsau, 37
arsau_analyze, 38
arsau_datasets, 39
ARSAU_KINDS, 40
ARSAU_REGISTRY, 40, *136*
ARSAU_YEARS, 40
assess_calibration, 41
assess_discrimination, 41
audit_variables, 42
auto_test, 42

bartlett_test, 43
benjamini_hochberg, 43
benjamini_yekutieli, 44
bh *(benjamini_hochberg)*, 43
bias_adjusted_estimate, 44
block_bootstrap, 45
bonferroni, 46
bootstrap, 46
bootstrap_632, 47
bootstrap_effect_size_ci, 48
bootstrap_validate, 49
bq_table_download, *536*
buttbp, 49
buttbp(), *713, 714*
buttbs, 50
butthp, 51
buttlp, 52
by_fdr *(benjamini_yekutieli)*, 44

cauchy_combination, 52
CCRSO_TABLE1_RELEASES, 53
CCRSO_TABLE2_FLOW, 53
CCRSO_TABLE3_AGE, 54
cepst, 54
cheatsheet, 55
check_referential_integrity, 55
chi2_goodness_of_fit, 56
chi2_independence, 56
ckan_metadata, 57
cles, 57
cliffs_delta, 58
cobalt::bal.tab, *601, 602*
cobalt::love.plot, *601*, *612*
cochrans_q, 58
coef.crr, 59
coefficient_of_variation, 59
cohens_f, 60
cohens_kappa, 60

cohens_w, 61
collinearity_diagnostics, 61
column_rule, 62
commands_by_category, 63, *852*
compute_goodness_of_fit, 63
compute_influence, 64
compute_residuals, 64
compute_vif, 65
correlation_matrix, 65
correlation_table, 66
create_reproducibility_manifest, 66
cross_validate, 67

d_to_nnt, 68
d_to_or, 68
d_to_r, 69
dagostino_pearson, 69
dataset_catalog, 70
decision_curve_analysis, 71
decoupling_test, 71
delete_d_jackknife, 72
designmatch::cardmatch, *603*
detect_overfitting, 73
dfa, 73
doob_trends, 74

e_value, 75
e_value_d, 75, *776*
e_value_hr, 76, *776*
e_value_or, 76, *776*
e_value_rr, 77, *776*
ecgdet, 77
effect_size_result, 78
effect_sizes, 79
effects, 79
epsilon_squared, 80
estimate_ate, 81
estimate_ate_gcomputation, 81
estimate_pi0, 82
estimate_pliv, 82
estimate_plr, 83
explain_file, 84
explain_known_files, 85
extenders_nonparam, 85
extenders_rdd, 85
extenders_spatial, 86
extenders_stats, 86
external_validate, 86

fairness_cityprofile, 87
fairness_metrics, 87, *88*
fairness_temporal, 88
fallback_procedure, 88
fisher_combined, 89
fisher_exact_test, 89
fixed_effects_meta, 90
fixed_sequence, 90
format_dataframe, 91
format_number, 91
friedman_test, 92
frns_metrics, 92
frns_predpol, 93
frns_temporal, 94
full_diagnostics, 94

gam_smoother, 95, *775*
get_cansim, *550*
glass_delta, 96
glm, *1001*
glm.fit, *1015*

harmonic_mean_p, 96
hazard_ratio_table, 97
hcepst, 97
hdecon, 98
hfd, 99
hierarchical_bonferroni, 100
hochberg, 100
holm, 101
holm_sidak, 101
hommel, 102
hosmer_lemeshow_test, 102
hrvfd, 103
hrvnl, 103
hrvtd, 104

i_squared, 105
incidence_rate_difference, 105
intraclass_correlation, 106

jackknife, 106

kde, 107, *775*
kendall_correlation, 108
kernel-codes, 108
KERNEL_BIWEIGHT *(kernel-codes)*, 108
kernel_cond_moments, 109, *775*
KERNEL_EPANECHNIKOV *(kernel-codes)*, 108

kernel_eval, 109, *775*
KERNEL_GAUSSIAN *(kernel-codes)*, 108
KERNEL_TRIANGULAR *(kernel-codes)*, 108
KERNEL_UNIFORM *(kernel-codes)*, 108
kfd, 110
ks_test_one_sample, 111
ks_test_two_sample, 111

leave_one_out_cv, 112
levene_test, 112
license_check, 113
likelihood_ratio_test, 113
lilliefors_test, 114
link_test, 114
local_fdr, 115
local_linear, 115, *775*
longitudinal_sim, 116
loocv_bandwidth, 117, *775*

mann_whitney_u, 117
manski_bounds, 118
mcnemar_test, 118
midranks, 119
model_comparison_table, 119
morie_anominate_ideal_points, 120
morie_anova_one_way, 121
morie_arch_in_mean, 121
morie_arsau_analyze_aggregate_summary, *39*, 122, *973*
morie_arsau_analyze_detailed_dataset, *39*, 123, *125*, *973*
morie_arsau_analyze_individual_records, *39*, 124, *973*
morie_arsau_analyze_main_records, *39*, 125, *973*
morie_arsau_analyze_probe_cycle_records, *39*, 126, *973*
morie_arsau_analyze_weapon_records, *39*, 126, *973*
morie_arsau_available_datasets, 127
morie_arsau_available_years, 128
morie_arsau_ckan_url, *39*, 128
morie_arsau_describe, 129
morie_arsau_download, 129
morie_arsau_fetch_sidecar, *39*, 130
morie_arsau_load_aggregate_summary, 131
morie_arsau_load_detailed_dataset, 131
morie_arsau_load_individual_records, 132
morie_arsau_load_main_records, 132
morie_arsau_load_probe_cycle_records, 133
morie_arsau_load_weapon_records, *127*, 133
morie_arsau_read_markdown_dictionary, *39*, 134
morie_arsau_read_sidecar, *130*, 134, *136*
morie_arsau_read_xlsx_dictionary, *39*, 135
morie_arsau_registry_df, *39*, 136
morie_arsau_sidecar_schema, *39*, *130*, 136
morie_arsau_sidecar_to_frame, *39*, *130*, 137
morie_audit_all_variables, *42*, 137
morie_audit_arsau_variables, *42*, 138
morie_audit_otis_variables, *42*, 138, *138*
morie_audit_public_outputs, 139
morie_audit_public_outputs(), *855*
morie_bayes_cpi_genomic, 140
morie_bayes_ridge_gibbs, 141
morie_bayesian_lasso_full, 142
morie_bayesian_ridge_regression, 143
morie_boot_basic_ci, 143
morie_boot_basic_ci(), *47*, *144*, *783*
morie_boot_run, 144
morie_boot_run(), *47*, *144*
morie_bootstrap_sample, 145
morie_build_outputs_manifest, 145
morie_builtin_db, 146
morie_cache_clear, 147, *148*, *503*, *505*, *797*
morie_cache_dir, 147, *147*, *503*, *505*, *551*, *692*, *797*
morie_cache_file, 149
morie_cache_list, 149
morie_cache_load, 150
morie_cache_store, 151
morie_calculate_ebac, 151
morie_calculate_ebac(), *557*
morie_calculate_ipw_weights, 152
morie_calibration_weights, 153
morie_canonicalize_cpads_data, 154
morie_causal_impact, 155

morie_causal_robust_se, 156
morie_causal_weighting, 157
morie_check_plugin_license, 158
morie_chi_square_test, 159
morie_ckan_search, 159, *500*, *595*
morie_classify_variable, *42*, 160
morie_cli_dump_catalog, 161
morie_cluster_sample, 161
morie_cnn_genomic, 162
morie_coherence, 163
morie_coin_independence, 163
morie_coin_oneway, 164
morie_coin_wilcoxon, 165
morie_compare_nested_logistic_models, 165
morie_compute_design_weights, 166
morie_concordance_incomplete, 167
morie_confusion_matrix_metrics, 167
morie_contingency_coefficient, 168
morie_control_comparison, 169
morie_control_median_test, 169
morie_copula_fit, 170
morie_copula_sample, 171
morie_corrections_uof_resource_ids, 171, *874*
morie_cpads_canonicalize_frame, 172
morie_cpads_contract, 173
morie_cpads_contract(), *230*
morie_cpads_has_raw_columns, 173
morie_cpads_infer_file_format, 174
morie_cpads_missing_variables, 174
morie_cpads_validate_frame, 175
morie_crypto_chacha20_poly1305_decrypt, 176
morie_crypto_chacha20_poly1305_encrypt, 176
morie_crypto_chacha20_poly1305_encrypt(), *176*
morie_crypto_hkdf_sha256, 177
morie_crypto_hybrid_decrypt, 178
morie_crypto_hybrid_encrypt, 178
morie_crypto_hybrid_keygen, 179
morie_crypto_keystore_create, 179
morie_crypto_keystore_list, 180
morie_crypto_keystore_load, 180
morie_crypto_keystore_store, 181
morie_crypto_liboqs_available, 181
morie_crypto_liboqs_version, 182
morie_crypto_mldsa65_keygen, 182
morie_crypto_mldsa65_sign, 183
morie_crypto_mldsa65_sign(), *183*
morie_crypto_mldsa65_verify, 183
morie_crypto_mlkem768_decaps, 184
morie_crypto_mlkem768_encaps, 184
morie_crypto_mlkem768_keygen, 185
morie_crypto_random_bytes, 185
morie_crypto_sodium_available, 186
morie_crypto_sodium_version, 186
morie_dataset_catalog, 187
morie_dataset_column_profile, 187
morie_dataset_detect_role, 188
morie_dataset_infer_level, 189
morie_dataset_info, 190
morie_dataset_load, 190
morie_dataset_portal_catalog, 191
morie_dataset_portal_catalog(), *192*, *201*, *202*, *234*
morie_dataset_portal_catalog_clear_cache, 192
morie_dataset_portal_catalog_clear_cache(), *191*
morie_dataset_profile, 192
morie_dataset_profile_summary_table, 193
morie_dataset_profile_to_list, 194
morie_dataset_suggest_plan, 194
morie_dataset_summarize_column, 195
morie_datasets_arcgis_item_by_id, 195
morie_datasets_arcgis_item_by_id(), *290*
morie_datasets_arcgis_item_metadata, 196
morie_datasets_arsau_aggregate_summary, 197
morie_datasets_arsau_detailed_dataset, 197
morie_datasets_arsau_uof_individual_records, 198
morie_datasets_arsau_uof_main_records, 198
morie_datasets_arsau_uof_probe_cycle_records, 199
morie_datasets_arsau_uof_weapon_records, 200
morie_datasets_bigquery, 201
morie_datasets_browse, 201

morie_datasets_browse(), *191*
morie_datasets_calgary_community_crime_stats, 203
morie_datasets_calgary_fire_response_calls *(morie_datasets_calgary_community_crime_stats)*, 203
morie_datasets_calgary_fire_stations *(morie_datasets_calgary_community_crime_stats)*, 203
morie_datasets_calgary_open_crime_adjacent_layers, 203
morie_datasets_calgary_opendata_bulk_layers *(morie_datasets_nyc_opendata_bulk_layers)*, 257
morie_datasets_calgary_socrata_by_id, 204
morie_datasets_chicago_arrests, 204
morie_datasets_chicago_community_areas, 206
morie_datasets_chicago_community_areas(), *208*, *213*
morie_datasets_chicago_crime, 207
morie_datasets_chicago_crime(), *206*, *211*, *212*, *214*, *215*, *221*
morie_datasets_chicago_crime_map, 209
morie_datasets_chicago_crime_map(), *208*, *211*, *213*
morie_datasets_chicago_crime_odata, 210
morie_datasets_chicago_crime_resolved, 212
morie_datasets_chicago_crime_soql, 214
morie_datasets_chicago_crime_soql(), *211*
morie_datasets_chicago_iucr_codes, 215
morie_datasets_chicago_iucr_codes(), *208*, *213*
morie_datasets_chicago_neighborhoods, 216
morie_datasets_chicago_opendata_bulk_layers *(morie_datasets_nyc_opendata_bulk_layers)*, 257
morie_datasets_chicago_police_beats, 218
morie_datasets_chicago_police_beats(), *208*, *213*
morie_datasets_chicago_police_districts, 219
morie_datasets_chicago_police_districts(), *208*, *213*
morie_datasets_chicago_socrata_by_id *(morie_datasets_nyc_socrata_by_id)*, 259
morie_datasets_chicago_wards, 221
morie_datasets_chicago_wards(), *208*, *213*
morie_datasets_ckan_package, 222
morie_datasets_ckan_search, 223
morie_datasets_corrections_uof_ethnic_origin, 223
morie_datasets_corrections_uof_incident_type, 224
morie_datasets_corrections_uof_incidents, 224, *874*
morie_datasets_corrections_uof_indigenous, 225
morie_datasets_corrections_uof_inmate_incident, 225
morie_datasets_corrections_uof_inmate_participant, 226
morie_datasets_corrections_uof_institution_summary, 226
morie_datasets_corrections_uof_location_summary, 227
morie_datasets_corrections_uof_race, 227
morie_datasets_corrections_uof_religion, 228
morie_datasets_corrections_uof_select_incident_summary, 228
morie_datasets_corrections_uof_staff_incident, 229
morie_datasets_cpads, 229
morie_datasets_cpd_public_arrests, 231
morie_datasets_edmonton_fire_stations *(morie_datasets_edmonton_police_stations)*, 232
morie_datasets_edmonton_open_crime_adjacent_layers *(morie_datasets_calgary_open_crime_adjacent_laye* 203
morie_datasets_edmonton_opendata_bulk_layers *(morie_datasets_nyc_opendata_bulk_layers)*, 257

morie_datasets_edmonton_police_stations, 232
morie_datasets_edmonton_socrata_by_id (*morie_datasets_calgary_socrata_by_id*), 204
morie_datasets_external_socrata_layers, 233
morie_datasets_external_socrata_layers(), *299*
morie_datasets_load_by_key, 234
morie_datasets_load_by_key(), *191*, *230*, *263*
morie_datasets_montreal_ckan_resource, 235
morie_datasets_montreal_justice_safety_layers, 236
morie_datasets_montreal_opendata_bulk_layers (*morie_datasets_nyc_opendata_bulk_layers*), 257
morie_datasets_montreal_sim_intervention_types, 237
morie_datasets_montreal_sim_interventions, 237
morie_datasets_namus_missing_persons, 238
morie_datasets_nibrs, 239
morie_datasets_nist_rds, 240
morie_datasets_nyc_boroughs, 240
morie_datasets_nyc_boroughs(), *246*, *252*
morie_datasets_nyc_boundaries_catalog, 241
morie_datasets_nyc_community_districts, 242
morie_datasets_nyc_council_districts, 243
morie_datasets_nyc_ntas_2020, 243
morie_datasets_nyc_nypd_arrests_historic, 244
morie_datasets_nyc_nypd_arrests_ytd, 245
morie_datasets_nyc_nypd_boro_crosswalk, 246
morie_datasets_nyc_nypd_by_key, 246
morie_datasets_nyc_nypd_by_key(), *252*
morie_datasets_nyc_nypd_complaint_historic, 247
morie_datasets_nyc_nypd_complaint_ytd, 248
morie_datasets_nyc_nypd_hate_crimes, 249
morie_datasets_nyc_nypd_law_books, 250
morie_datasets_nyc_nypd_layers, 251
morie_datasets_nyc_nypd_layers(), *247*, *253*
morie_datasets_nyc_nypd_offense_codes, 251
morie_datasets_nyc_nypd_resolved, 252
morie_datasets_nyc_nypd_resolved(), *246*
morie_datasets_nyc_nypd_uof_incidents, 254
morie_datasets_nyc_nypd_uof_subjects, 255
morie_datasets_nyc_nypd_vehicle_stops, 256
morie_datasets_nyc_opendata_bulk_layers, 257
morie_datasets_nyc_police_precincts, 258
morie_datasets_nyc_police_precincts(), *252*
morie_datasets_nyc_school_districts, 259
morie_datasets_nyc_socrata_by_id, 259
morie_datasets_nyc_stop_and_frisk, 260
morie_datasets_nyc_zctas, 261
morie_datasets_ontario_ckan_by_key, 261
morie_datasets_ontario_ckan_by_key(), *262*
morie_datasets_ontario_ckan_layers, 262
morie_datasets_ontario_ckan_layers(), *233*, *261*, *299*
morie_datasets_otis_a01, 262, *875*
morie_datasets_otis_a01_restrictive_confinement, 263
morie_datasets_otis_a01_restrictive_confinement(), *262*, *263*
morie_datasets_otis_b01_segregation_detailed, 264
morie_datasets_otis_b02_segregation_total_days, 264

morie_datasets_otis_b03_seg_alerts_by_institution, 265
morie_datasets_otis_b04_seg_consecutive_by_region, 266
morie_datasets_otis_b05_seg_consecutive_lengths, 266
morie_datasets_otis_b06_seg_reason_by_institution, 267
morie_datasets_otis_b07_seg_alerts_by_gender, 268
morie_datasets_otis_b08_seg_consecutive_by_institution, 268
morie_datasets_otis_b09_seg_n_times, 269
morie_datasets_otis_c01_individuals_total, 270
morie_datasets_otis_c02_individuals_by_institution, 270
morie_datasets_otis_c03_individuals_race_by_gender, 271
morie_datasets_otis_c04_individuals_race_by_region, 272
morie_datasets_otis_c05_individuals_religion_by_region, 272
morie_datasets_otis_c06_individuals_age_by_region, 273
morie_datasets_otis_c07_individuals_alerts, 274
morie_datasets_otis_c08_individuals_religion_by_gender, 274
morie_datasets_otis_c09_individuals_age_by_gender, 275
morie_datasets_otis_c10_aggregate_durations_by_institution, 276
morie_datasets_otis_c11_aggregate_lengths, 276
morie_datasets_otis_c12_aggregate_durations_by_region, 277
morie_datasets_otis_d01_deaths_in_custody, 278
morie_datasets_otis_d02_deaths_by_gender, 279
morie_datasets_otis_d03_deaths_by_race, 279
morie_datasets_otis_d04_deaths_by_religion, 280
morie_datasets_otis_d05_deaths_by_age_category, 281
morie_datasets_otis_d06_cause_by_alert, 281
morie_datasets_otis_d07_alerts_by_housing_unit, 282
morie_datasets_ottawa_open_crime_adjacent_layers *(morie_datasets_calgary_open_crime_adjacent_laye*, 203
morie_datasets_ottawa_opendata_bulk_layers *(morie_datasets_nyc_opendata_bulk_layers)*, 257
morie_datasets_siu_director_reports, 283
morie_datasets_siu_report_fields, 283
morie_datasets_siu_report_text, 284
morie_datasets_statcan_ccjs_cubes, 284
morie_datasets_statcan_cube_metadata, 285
morie_datasets_statcan_cube_metadata(), *235*
morie_datasets_statcan_full_csv_url, 285
morie_datasets_statcan_vectors, 286
morie_datasets_summary, 287
morie_datasets_toronto_ambulance_stations, 287
morie_datasets_toronto_asr_miscellaneous, 288
morie_datasets_toronto_open_ckan_resource, 288
morie_datasets_toronto_open_ckan_resource(), *289*
morie_datasets_toronto_open_crime_adjacent_layers, 289
morie_datasets_toronto_opendata_bulk_layers *(morie_datasets_nyc_opendata_bulk_layers)*, 257
morie_datasets_toronto_zoning_per_neighbourhood, *196*, 289
morie_datasets_tps_2008_firs, 290
morie_datasets_tps_2009_firs, 291
morie_datasets_tps_2010_firs, 292
morie_datasets_tps_2011_firs, 293
morie_datasets_tps_2012_firs, 294
morie_datasets_tps_2013_firs, 295
morie_datasets_tps_administrative, 296
morie_datasets_tps_arcgis_hub_by_id,

297
morie_datasets_tps_arcgis_hub_by_id(), *195*, *235*, *290–297*, *301–325*, *328*, *329*, *332–344*, *346–349*, *351*, *353–365*
morie_datasets_tps_arcgis_hub_download, 299
morie_datasets_tps_arcgis_hub_layers, *196*, 299
morie_datasets_tps_arcgis_hub_layers(), *297*, *298*
morie_datasets_tps_arrested_and_charged_persons, 301
morie_datasets_tps_arrests_and_strip_searches, 302
morie_datasets_tps_assault, 303
morie_datasets_tps_automobile_ksi, 303
morie_datasets_tps_autotheft, 304
morie_datasets_tps_bicycle_thefts, 305
morie_datasets_tps_bicycletheft, 306
morie_datasets_tps_breakandenter, 307
morie_datasets_tps_budget_2020, 307
morie_datasets_tps_budget_2021, 308
morie_datasets_tps_budget_2022, 309
morie_datasets_tps_budget_2023, 310
morie_datasets_tps_budget_2024, 311
morie_datasets_tps_budget_2025, 312
morie_datasets_tps_budget_2026, 313
morie_datasets_tps_budget_by_command, 314
morie_datasets_tps_calls_for_service_attended, 315
morie_datasets_tps_community_safety_indicators, 316
morie_datasets_tps_complaint_dispositions, 317
morie_datasets_tps_cyclist_ksi, 318
morie_datasets_tps_dispatched_calls_by_division, 319
morie_datasets_tps_facilities, 320
morie_datasets_tps_fatals_ksi, 321
morie_datasets_tps_firearms_top_calibres, 322
morie_datasets_tps_gross_expenditures_by_division, 323
morie_datasets_tps_gross_operating_budget, 324
morie_datasets_tps_hatecrimes, 325
morie_datasets_tps_homicide, 326
morie_datasets_tps_homicides, 326
morie_datasets_tps_intimate_partner_family_violence, 327
morie_datasets_tps_investigated_alleged_misconduct, 328
morie_datasets_tps_killed_and_seriously_injured, 329
morie_datasets_tps_layers, 330
morie_datasets_tps_major_crime, 330
morie_datasets_tps_mha_apprehensions, 331
morie_datasets_tps_miscellaneous_calls_for_service, 332
morie_datasets_tps_miscellaneous_data, 333
morie_datasets_tps_miscellaneous_firearms, 334
morie_datasets_tps_motorcylist_ksi, 335
morie_datasets_tps_neighbourhood_crime_rates, 336
morie_datasets_tps_passenger_ksi, 337
morie_datasets_tps_patrol_zone, 338
morie_datasets_tps_pedestrian_ksi, 339
morie_datasets_tps_personnel_by_command, 340
morie_datasets_tps_personnel_by_rank, 341
morie_datasets_tps_personnel_by_rank_by_division, 342
morie_datasets_tps_persons_in_crisis_calls_for_service, 343
morie_datasets_tps_police_divisions, 344
morie_datasets_tps_police_divisions(), *345*
morie_datasets_tps_psdp_resolved, 345
morie_datasets_tps_regulated_interactions, 346
morie_datasets_tps_reported_crimes, 347
morie_datasets_tps_robbery, 348
morie_datasets_tps_search_of_persons, 348

morie_datasets_tps_shooting_firearm_discharges, 349
morie_datasets_tps_shootings, 350
morie_datasets_tps_staffing_by_command, 351
morie_datasets_tps_theft_from_motor_vehicle, 352
morie_datasets_tps_theft_over, 352
morie_datasets_tps_tickets_issued, 353
morie_datasets_tps_top_20_offences_of_firearm_seizures, 354
morie_datasets_tps_total_public_complaints, 355
morie_datasets_tps_traffic_collisions, 356
morie_datasets_tps_use_of_force_call_for_service_types, 357
morie_datasets_tps_use_of_force_call_sources_by_month, 358
morie_datasets_tps_use_of_force_gender_composition, 359
morie_datasets_tps_use_of_force_location_of_occurrences, 360
morie_datasets_tps_use_of_force_occurrence_category, 361
morie_datasets_tps_use_of_force_time_of_day_trends, 362
morie_datasets_tps_use_of_force_use_of_force_types_and_perceived_weapons, 363
morie_datasets_tps_victims_of_crime, 364
morie_datasets_vancouver_community_centres *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_community_food_markets *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_disability_parking *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_fire_halls *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_graffiti *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_homeless_shelters *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_noise_control_areas *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_opendata_bulk_layers *(morie_datasets_nyc_opendata_bulk_layers)*, 257
morie_datasets_vancouver_opendata_by_id, 365
morie_datasets_vancouver_opendata_by_id(), *235*
morie_datasets_vancouver_opendata_layers, 366
morie_datasets_vancouver_opendata_layers(), *365*
morie_datasets_vancouver_property_use_inspection_distri *(vancouver_crime_adjacent)*, 1090
morie_datasets_vancouver_public_art *(vancouver_crime_adjacent)*, 1090
morie_datasets_vpd_crime, 366, *953*
morie_datasets_vpd_crime(), *954*, *1090*
morie_datasets_vpd_legal_disclaimer, 369
morie_datasets_vpd_legal_disclaimer(), *367*
morie_db_connect, 369
morie_db_create_indexes, 370
morie_dbscan_clustering, 371
morie_dcc_multivariate_garch, 371
morie_decision_tree_split, 372
morie_deep_learning_genomic, 373
morie_default_synthetic_name_map, 374
morie_default_synthetic_name_map(), *509*
morie_default_workflow_map, 374
morie_desc_atkinson, 375
morie_desc_cramers_v, 375
morie_desc_gini, 376
morie_desc_kappa, 376
morie_desc_winsorize, 377
morie_describe, 377, *378*
morie_describe_by_name, 378, *378*
morie_design_effect, 379
morie_det_rng, 379
morie_det_rng(), *140–142*, *162*, *373*, *513*,

515, *518*, *525*, *527*, *735*, *737*, *764*, *944*, *945*, *970*
morie_det_rng_sha_hex, 380
morie_did_2x2, 381, *384*, *385*, *387*, *389*, *392*, *395*, *400*, *401*, *404*
morie_did_aggregate_gt_att, 382, *397*
morie_did_bacon_decomposition, 383
morie_did_chaisemartin_dhaultfoeuille, 383, *389*, *403*
morie_did_continuous_treatment, 384
morie_did_diagnostics, 385
morie_did_doubly_robust, *381*, 386
morie_did_event_study, 387
morie_did_fuzzy, 388
morie_did_group_time_att, *381*, *382*, 389, *397*
morie_did_heterogeneous, 390
morie_did_panel_fe, 391, *403*
morie_did_parallel_trends_data, 392
morie_did_placebo_test_group, 392
morie_did_placebo_test_outcome, 393
morie_did_placebo_test_time, 394
morie_did_repeated_cross_section, 395
morie_did_sensitivity_analysis, 396
morie_did_staggered, 397
morie_did_synthdid_estimate, 398
morie_did_synthetic, *398*, 399, *399*
morie_did_test_parallel_trends, 400
morie_did_triple_difference, 401
morie_did_twoway_fe_weights, 402
morie_did_wild_cluster_bootstrap, 403
morie_did_wild_cluster_bootstrap(), *1096*
morie_diffu_diffusion_forward, 404
morie_download_bootstrap, 405
morie_dp_gaussian_mixture, 406
morie_dsp_acf_from_psd, 407
morie_dsp_alpha_trimmed_mean, 407
morie_dsp_amplitude_histogram, 408
morie_dsp_arc_length, 408
morie_dsp_band_power, 409
morie_dsp_baseline_correlation, 409
morie_dsp_centroidal_time, 410
morie_dsp_coherence, 410
morie_dsp_coherence_spectrum, 411
morie_dsp_comb, 412
morie_dsp_complex_cepstrum, 412
morie_dsp_complex_demodulation, 413
morie_dsp_crest_factor, 414
morie_dsp_cross_correlation, 414
morie_dsp_csd, 415
morie_dsp_cv, 415
morie_dsp_derivative_detect, 416
morie_dsp_dicrotic_notch, 417
morie_dsp_ensemble_average, 417
morie_dsp_entropy_histogram, 418
morie_dsp_even_odd, 418
morie_dsp_fbm_synthesis, 419
morie_dsp_form_factor, 419
morie_dsp_fractal_dim_psd, 420
morie_dsp_hann_filter, 420
morie_dsp_higuchi_fd, 421
morie_dsp_hilbert_envelope, 422
morie_dsp_hjorth, 422
morie_dsp_hjorth_activity, 423
morie_dsp_hjorth_complexity, 423
morie_dsp_hjorth_mobility, 424
morie_dsp_homomorphic, 424
morie_dsp_hr_from_rr, 425
morie_dsp_integrated_emg, 425
morie_dsp_katz_fd, 426
morie_dsp_lms, 426
morie_dsp_matched, 427
morie_dsp_mean_abs, 427
morie_dsp_mean_frequency, 428
morie_dsp_median_filter, 428
morie_dsp_median_frequency, 429
morie_dsp_min_phase, 430
morie_dsp_moving_average, 430
morie_dsp_myopulse_rate, 431
morie_dsp_nlms, 431
morie_dsp_notch, 432
morie_dsp_onset_detect, 433
morie_dsp_pan_tompkins, 433
morie_dsp_parzen_pdf, 434
morie_dsp_psd_bartlett, 435
morie_dsp_psd_periodogram, 435
morie_dsp_psd_to_db, 436
morie_dsp_psd_welch, 436
morie_dsp_qrs_features, 437
morie_dsp_rls, 438
morie_dsp_rms, 438
morie_dsp_ruler_fd, 439
morie_dsp_shannon_energy, 439
morie_dsp_shape_factor, 440
morie_dsp_slope_sign_changes, 440

morie_dsp_snr, 441
morie_dsp_snr_improvement, 442
morie_dsp_spectral_edge, 442
morie_dsp_spectral_entropy, 443
morie_dsp_spectral_flatness, 443
morie_dsp_spectral_kurtosis, 444
morie_dsp_spectral_moment, 444
morie_dsp_spectral_ratio, 445
morie_dsp_synchronized_average, 446
morie_dsp_t_wave, 446
morie_dsp_teager_energy, 447
morie_dsp_template_match, 447
morie_dsp_threshold_detect, 448
morie_dsp_turning_points, 449
morie_dsp_turns_count, 449
morie_dsp_variance_ratio, 450
morie_dsp_waveform_length, 450
morie_dsp_waveform_length_norm, 451
morie_dsp_wiener_filter, 451
morie_dsp_wiener_hopf, 452
morie_dsp_willison_amplitude, 452
morie_dsp_window, 453
morie_dsp_zero_crossing, 454
morie_e_value, 454
morie_effects_comparisons, 455
morie_effects_emmeans, 456
morie_effects_predictions, 456
morie_effects_slopes, 457
morie_effects_tidy, 457
morie_eg_coint, 458
morie_egarch_model, 458
morie_ensure_extras, 459
morie_entheo_analyze_subject, 460
morie_entheo_available_subjects, 461
morie_entheo_clone_dmt_imaging, 461
morie_entheo_dataset_overview, 462
morie_entheo_dynamic_functional_connectivity, 463
morie_entheo_load_eeg_region, 463
morie_entheo_load_fmri_subject, 464
morie_entheo_lz_complexity, 464
morie_entheo_spectral_band_power, 465
morie_estimate_aipw, 466
morie_estimate_atc, 467
morie_estimate_ate, 468
morie_estimate_ate(), *771*
morie_estimate_att, 469
morie_estimate_cate, 469
morie_estimate_double_ml, 471
morie_estimate_g_computation, 472
morie_estimate_gate, 473
morie_estimate_irm, 474
morie_estimate_late, 476
morie_estimate_propensity_scores, 477
morie_ewma_volatility, 478
morie_fairness_apply_profile, *87*, 478
morie_fairness_average_odds_difference, *87*, 479
morie_fairness_bias_amplification, *87*, 480
MORIE_FAIRNESS_CANONICAL_FIELDS, 481
morie_fairness_city_profile, *87*, 482
morie_fairness_column_map, 483
morie_fairness_demographic_parity, *87*, 483
morie_fairness_disparate_impact, *87*, 484
morie_fairness_equalized_odds, *87*, 485
morie_fairness_get_city, *87*, *482*, 487
morie_fairness_gini, *87*, 487
morie_fairness_list_cities, *87*, 488
morie_fairness_noisy_or_detection, 488
morie_fairness_predpol, 489
morie_fairness_predpol_aggregate_areas, 489
morie_fairness_predpol_calibration_audit, 490
morie_fairness_predpol_score_disparity, 491
morie_fairness_predpol_temporal_audit, 491
morie_fairness_register_city, *87*, 492
morie_fairness_simulate_biased_crime_data, 493
morie_fairness_simulation, 494
morie_fairness_xai, 494
morie_fairness_xai_ale, 494
morie_fairness_xai_ceteris_paribus, 495
morie_fairness_xai_partial_dependence, 496
morie_fairness_xai_permutation_importance, 496
morie_fairness_xai_shap_values, 497
morie_fast_available, 498

morie_fdr_qvalues, 499
morie_fetch, *159, 160*, 499, *501*, *595*
morie_fetch_arcgis, *500*, 501
morie_fetch_ckan, *159, 160*, 502
morie_fetch_siu, 503, *786*, *796, 797*
morie_fetch_tps, 504
morie_fisher_exact_test, 505
morie_garch_fit, 506
morie_gblup_full, 507
morie_generate_ar_coefficients, 507
morie_generate_synthetic_data, 508
morie_generate_synthetic_data(), *374*
morie_generate_var_coefficients, 509
morie_genomic_cross_validation, 510
morie_geostat_krige, 510
morie_geostat_variogram, 511
morie_ghosal_adaptation, 512
morie_ghosal_bernstein_von_mises, 513
morie_ghosal_contraction_rate, 514
morie_ghosal_dirichlet_posterior, 514
morie_ghosal_dpmixture_density, 515
morie_ghosal_empirical_bayes, 516
morie_ghosal_gp_matern, 516
morie_ghosal_gp_squared_exponential, 517
morie_ghosal_hierarchical_bayes, 518
morie_ghosal_log_density, 519
morie_ghosal_moment_matching, 519
morie_ghosal_neutral_right, 520
morie_ghosal_np_classification, 521
morie_ghosal_np_regression, 522
morie_ghosal_np_testing, 522
morie_ghosal_posterior_consistency, 523
morie_ghosal_sieve_prior, 524
morie_ghosal_stick_breaking_trunc, 524
morie_ghosal_survival_beta_process, 525
morie_ghosal_wavelet_prior, 526
morie_gpl_compatible_licenses, 526
morie_gradient_boosting_ensemble, 527
morie_gradient_boosting_genomic, 528
morie_gradient_descent_vanilla, 529
morie_grid_search_cv, 529
morie_grm_vanraden, 530
morie_gxe_interaction_model, 531
morie_hawkes_fit, 532
morie_hurst_r, 533
morie_infer_measurement_level, 533
morie_ingest_bigquery_build_sql, 534, *536*
morie_ingest_bigquery_query, *534*, 535, *536*, *537*
morie_ingest_bigquery_table, 536, *536*, *538–540*
morie_ingest_chicago_crime, 538, *539*, *540*
morie_ingest_chicago_crime_bigquery, 539
morie_ingest_chicago_resources, 540
morie_ingest_chicago_socrata, *538*, 540
morie_ingest_cihi_xlsx, 541
morie_ingest_ckan_fetch_package_csvs, 542
morie_ingest_ckan_package_search, 543, *546*
morie_ingest_ckan_package_show, 544
morie_ingest_ckan_read_resource, 544
morie_ingest_ckan_resource_show, *544*, 545
morie_ingest_ckan_search_packages, *543*, 546
morie_ingest_forensics_namus_missing, 547
morie_ingest_forensics_nibrs, 548
morie_ingest_forensics_nist_rds, 549
morie_ingest_statcan_cansim, 550, *551*
morie_ingest_statcan_csv, *550*, 551
morie_ingest_tps_feature_layer, 552, *553*
morie_ingest_tps_fetch, 553
morie_ingest_tps_layers, *552, 553*, 554
morie_inspect_output, 554
morie_install_extras, 555
morie_install_extras(), *459*
morie_is_over_legal_limit, 557
morie_iv_anderson_rubin, 557
morie_iv_anderson_rubin_ci, 558
morie_iv_conditional_lr, 559
morie_iv_control_function, 560
morie_iv_cragg_donald, 560
morie_iv_cue_gmm, 561
morie_iv_diagnostics, 562
morie_iv_durbin_wu_hausman, 563
morie_iv_first_stage_diagnostics, 563

morie_iv_gmm, 564
morie_iv_hansen_j, 565
morie_iv_hausman, 565
morie_iv_jive, 566
morie_iv_kleibergen_paap, 566
morie_iv_liml, 567
morie_iv_panel, 568
morie_iv_probit, 568
morie_iv_residual_analysis, 569
morie_iv_sargan, 570
morie_iv_split_sample, 570
morie_iv_stock_yogo, 571
morie_iv_tsls, 571
morie_iv_wald, 572
morie_jackknife_estimate, 573
morie_johansen_cointegration, 574
morie_kalman_filter, 574
morie_kendall_tau, 575
morie_kendall_tau_partial, 576
morie_kernel_pca, 576
morie_kmeans_clustering, 577
morie_kruskal_wallis_test, 578
morie_laniyonu_actuarial_risk_disparity, 578
morie_laniyonu_gentrification_policing, 580
morie_laniyonu_smi_force_disparity, 582
morie_lcmm_latent_class, 584
morie_learning_curve, 585
morie_levene_test, 585
morie_license_metadata, 586
morie_linear_regression_ols, 586
morie_list_datasets, 587
morie_list_morie_modules, 588
morie_llm_agent_available, 588
morie_llm_ask, 589
morie_llm_ask_multi, 589
morie_llm_detect_provider, 590
morie_llm_list_freeapi_models, 591
morie_llm_probe_freeapi, 591
morie_llm_probe_ollama, 592
morie_llm_request_completion, 592
morie_load_cpads, 593
morie_load_cpads_data, 594
morie_load_dataset, *187*, 594
morie_locfdr_estimate, 595
morie_mann_whitney_test, 596
morie_marker_variance, 597
morie_matching_abadie_imbens_se, 598
morie_matching_atc_matched, 599
morie_matching_ate_matched, 599
morie_matching_att_matched, 600
morie_matching_balance, 601
morie_matching_balance_table, 602
morie_matching_cardinality, 603
morie_matching_cem, 604
morie_matching_common_support, 605
morie_matching_doubly_robust, 606
morie_matching_entropy_balance, 607
morie_matching_estimate_propensity, 608
morie_matching_exact, 609
morie_matching_full, 609
morie_matching_genetic, 610
morie_matching_longitudinal, 611
morie_matching_love_plot_data, 612
morie_matching_mahalanobis, 613
morie_matching_multi_treatment, 614
morie_matching_nearest_neighbor, 615
morie_matching_optimal_pair, 616
morie_matching_overlap, 617
morie_matching_quality, 617
morie_matching_rosenbaum_bounds, 618
morie_matching_subclassify, 619
morie_matching_trim_propensity, 620
morie_matching_variable_ratio, 621
morie_meta_rma, 622
morie_midas_regression, 623
morie_mini_batch_gradient, 623
morie_ml_apply_smote, 624
morie_ml_eval_robustness, 625
morie_multi_trait_gblup, 625
morie_multiple_testing, 626
morie_mvn_with_covariance, 627
morie_mvnorm_pmv, 628
morie_mvnorm_sample, 629
morie_nbeats_basis, 629
morie_np_kernel_reg, 630
morie_nyc_nypd_socrata_cap_note, 631
morie_odds_ratio_ci, 632
morie_omega_squared, 632
morie_one_sample_coverage, 633
morie_one_sample_t_test, 633
morie_ordered_alternatives_test, 634
morie_ordered_categories, 635

morie_otis_aipw_ate, 635
morie_otis_aipw_superlearner, 637
morie_otis_alert_state_combo *(morie_otis_primitives)*, 701
morie_otis_all_analyses, 637, *700*
morie_otis_all_analyze, 638
morie_otis_analyze_a01, 639
morie_otis_analyze_a01_dlrm *(morie_otis_analyze_a01_ruhela_formulations)*, 641
morie_otis_analyze_a01_mrm *(morie_otis_analyze_a01_ruhela_formulations)*, 641
morie_otis_analyze_a01_mrm_alt_age *(morie_otis_analyze_a01_ruhela_alt_age)*, 639
morie_otis_analyze_a01_mrm_alt_gender *(morie_otis_analyze_a01_ruhela_alt_gender)*, 640
morie_otis_analyze_a01_mrm_alt_toronto *(morie_otis_analyze_a01_ruhela_alt_toronto)*, 641
morie_otis_analyze_a01_mrm_per_year *(morie_otis_analyze_a01_ruhela_per_year)*, 642
morie_otis_analyze_a01_mrm_subgroup_female *(morie_otis_analyze_a01_ruhela_subgroup_female)*, 643
morie_otis_analyze_a01_mrm_subgroup_male *(morie_otis_analyze_a01_ruhela_subgroup_male)*, 643
morie_otis_analyze_a01_ruhela_alt_age, 639
morie_otis_analyze_a01_ruhela_alt_gender, 640
morie_otis_analyze_a01_ruhela_alt_toronto, 641
morie_otis_analyze_a01_ruhela_formulations, 641
morie_otis_analyze_a01_ruhela_per_year, 642
morie_otis_analyze_a01_ruhela_subgroup_female, 643
morie_otis_analyze_a01_ruhela_subgroup_male, 643
morie_otis_analyze_a01_with_csi_context, 644
morie_otis_analyze_all, 645
morie_otis_analyze_b01, 645
morie_otis_analyze_b01_dlrm *(morie_otis_analyze_b01_ruhela_formulations)*, 647
morie_otis_analyze_b01_mrm *(morie_otis_analyze_b01_ruhela_formulations)*, 647
morie_otis_analyze_b01_mrm_alt_age *(morie_otis_analyze_b01_ruhela_alt_age)*, 646
morie_otis_analyze_b01_mrm_alt_gender *(morie_otis_analyze_b01_ruhela_alt_gender)*, 646
morie_otis_analyze_b01_mrm_alt_toronto *(morie_otis_analyze_b01_ruhela_alt_toronto)*, 647
morie_otis_analyze_b01_mrm_per_year *(morie_otis_analyze_b01_ruhela_per_year)*, 648
morie_otis_analyze_b01_mrm_subgroup_female *(morie_otis_analyze_b01_ruhela_subgroup_female)*, 649
morie_otis_analyze_b01_mrm_subgroup_male *(morie_otis_analyze_b01_ruhela_subgroup_male)*, 649
morie_otis_analyze_b01_ruhela_alt_age, 646
morie_otis_analyze_b01_ruhela_alt_gender, 646
morie_otis_analyze_b01_ruhela_alt_toronto, 647
morie_otis_analyze_b01_ruhela_formulations, 647
morie_otis_analyze_b01_ruhela_per_year, 648
morie_otis_analyze_b01_ruhela_subgroup_female, 649
morie_otis_analyze_b01_ruhela_subgroup_male, 649
morie_otis_analyze_b02, 650
morie_otis_analyze_b02_dlrm *(morie_otis_analyze_b02_ruhela_formulations)*, 651
morie_otis_analyze_b02_mrm *(morie_otis_analyze_b02_ruhela_formulations)*, 651
morie_otis_analyze_b02_mrm_alt_age *(morie_otis_analyze_b02_ruhela_alt_age)*,

650
morie_otis_analyze_b02_mrm_alt_region *(morie_otis_analyze_b02_ruhela_alt_region)*, 651
morie_otis_analyze_b02_ruhela_alt_age, 650
morie_otis_analyze_b02_ruhela_alt_region, 651
morie_otis_analyze_b02_ruhela_formulations, 651
morie_otis_analyze_b03, 652
morie_otis_analyze_b03_mrm_aggregate *(morie_otis_analyze_b03_ruhela_aggregate)*, 652
morie_otis_analyze_b03_ruhela_aggregate, 652
morie_otis_analyze_b04, 653
morie_otis_analyze_b04_mrm_aggregate *(morie_otis_analyze_b04_ruhela_aggregate)*, 653
morie_otis_analyze_b04_ruhela_aggregate, 653
morie_otis_analyze_b05, 654
morie_otis_analyze_b05_mandela_classification, 654
morie_otis_analyze_b05_ruhela_aggregate, 655
morie_otis_analyze_b06, 655
morie_otis_analyze_b06_mrm_aggregate *(morie_otis_analyze_b06_ruhela_aggregate)*, 656
morie_otis_analyze_b06_ruhela_aggregate, 656
morie_otis_analyze_b07, 656
morie_otis_analyze_b07_mrm_aggregate *(morie_otis_analyze_b07_ruhela_aggregate)*, 657
morie_otis_analyze_b07_ruhela_aggregate, 657
morie_otis_analyze_b08, 657
morie_otis_analyze_b08_mrm_aggregate *(morie_otis_analyze_b08_ruhela_aggregate)*, 658
morie_otis_analyze_b08_ruhela_aggregate, 658
morie_otis_analyze_b09, 658
morie_otis_analyze_b09_mrm_aggregate *(morie_otis_analyze_b09_ruhela_aggregate)*, 659
morie_otis_analyze_b09_ruhela_aggregate, 659
morie_otis_analyze_c01, 660
morie_otis_analyze_c01_mrm_aggregate *(morie_otis_analyze_c01_ruhela_aggregate)*, 660
morie_otis_analyze_c01_mrm_aggregate_region_cluster *(morie_otis_analyze_c01_ruhela_aggregate_region_cluster)*, 661
morie_otis_analyze_c01_ruhela_aggregate, 660
morie_otis_analyze_c01_ruhela_aggregate_region_cluster, 661
morie_otis_analyze_c02, 662
morie_otis_analyze_c02_mrm_aggregate *(morie_otis_analyze_c02_ruhela_aggregate)*, 662
morie_otis_analyze_c02_ruhela_aggregate, 662
morie_otis_analyze_c03, 663
morie_otis_analyze_c03_mrm_aggregate *(morie_otis_analyze_c03_ruhela_aggregate)*, 663
morie_otis_analyze_c03_ruhela_aggregate, 663
morie_otis_analyze_c04, 664
morie_otis_analyze_c04_mrm_aggregate *(morie_otis_analyze_c04_ruhela_aggregate)*, 664
morie_otis_analyze_c04_mrm_aggregate_region_cluster *(morie_otis_analyze_c04_ruhela_aggregate_region_cluster)*, 665
morie_otis_analyze_c04_ruhela_aggregate, 664
morie_otis_analyze_c04_ruhela_aggregate_region_cluster, 665
morie_otis_analyze_c05, 665
morie_otis_analyze_c05_mrm_aggregate *(morie_otis_analyze_c05_ruhela_aggregate)*, 666
morie_otis_analyze_c05_ruhela_aggregate, 666
morie_otis_analyze_c06, 666
morie_otis_analyze_c06_mrm_aggregate *(morie_otis_analyze_c06_ruhela_aggregate)*, 667
morie_otis_analyze_c06_ruhela_aggregate,

667
morie_otis_analyze_c07, 667
morie_otis_analyze_c07_mrm_aggregate (*morie_otis_analyze_c07_ruhela_aggregate*), 668
morie_otis_analyze_c07_ruhela_aggregate, 668
morie_otis_analyze_c08, 668
morie_otis_analyze_c08_mrm_aggregate (*morie_otis_analyze_c08_ruhela_aggregate*), 669
morie_otis_analyze_c08_ruhela_aggregate, 669
morie_otis_analyze_c09, 669
morie_otis_analyze_c09_mrm_aggregate (*morie_otis_analyze_c09_ruhela_aggregate*), 670
morie_otis_analyze_c09_ruhela_aggregate, 670
morie_otis_analyze_c10, 670
morie_otis_analyze_c10_mrm_aggregate (*morie_otis_analyze_c10_ruhela_aggregate*), 671
morie_otis_analyze_c10_ruhela_aggregate, 671
morie_otis_analyze_c11, 671
morie_otis_analyze_c11_mandela_classification, 672
morie_otis_analyze_c11_mrm_aggregate (*morie_otis_analyze_c11_ruhela_aggregate*), 672
morie_otis_analyze_c11_ruhela_aggregate, 672
morie_otis_analyze_c12, 673
morie_otis_analyze_c12_mrm_aggregate (*morie_otis_analyze_c12_ruhela_aggregate*), 673
morie_otis_analyze_c12_ruhela_aggregate, 673
morie_otis_analyze_c_chi2, 659
morie_otis_analyze_d01, 674
morie_otis_analyze_d02, 675
morie_otis_analyze_d02_mrm_aggregate (*morie_otis_analyze_d02_ruhela_aggregate*), 675
morie_otis_analyze_d02_ruhela_aggregate, 675
morie_otis_analyze_d03, 676
morie_otis_analyze_d03_mrm_aggregate (*morie_otis_analyze_d03_ruhela_aggregate*), 676
morie_otis_analyze_d03_ruhela_aggregate, 676
morie_otis_analyze_d04, 677
morie_otis_analyze_d04_mrm_aggregate (*morie_otis_analyze_d04_ruhela_aggregate*), 677
morie_otis_analyze_d04_ruhela_aggregate, 677
morie_otis_analyze_d05, 678
morie_otis_analyze_d05_mrm_aggregate (*morie_otis_analyze_d05_ruhela_aggregate*), 678
morie_otis_analyze_d05_ruhela_aggregate, 678
morie_otis_analyze_d06, 679
morie_otis_analyze_d07, 679
morie_otis_analyze_d_chi2, 674
morie_otis_analyze_otis_mandela_provincial_vs_federal, 680
morie_otis_analyze_ruhela_grid, 680
morie_otis_analyze_ruhela_master, 681
morie_otis_analyze_ruhela_per_year, 682
morie_otis_analyzers, 683
morie_otis_astcmb, 683
morie_otis_causal_grid, 684
morie_otis_churn, 684
morie_otis_churn_analyze_all, 685
morie_otis_classify_mandela_combo, 686
morie_otis_descriptives (*morie_otis_primitives*), 701
morie_otis_disciplinary_medical_overlap, 687
morie_otis_dml (*morie_otis_primitives*), 701
morie_otis_embedding_distribution, 688
morie_otis_intra_year_transition_matrix, 688
morie_otis_ipw_ate, 689
morie_otis_irm_dml, 690
morie_otis_irr_glmm_vm, 691
morie_otis_load, 692, *700*, *701*
morie_otis_make_pair_a, 693

morie_otis_make_pair_alert_to_volatility_adj, 693
morie_otis_make_pair_alert_to_volatility_all, 694
morie_otis_make_pair_alert_to_volatility_naive, 695
morie_otis_make_pair_alert_to_volatility_rumelia, 695
morie_otis_make_pair_b, 696
morie_otis_make_pair_c, 697
morie_otis_mortification_cooccurrence, 697
morie_otis_otdesc, 698
morie_otis_otdml, 698
morie_otis_path_complexity_gini, 699
morie_otis_pipeline, 700
morie_otis_plr, 700
morie_otis_primitives, 701
morie_otis_psm, 702
morie_otis_psm_subclass, 703
morie_otis_rc_trends *(morie_otis_primitives)*, 701
morie_otis_rctrnd, 703
morie_otis_regC_demog_contingency, 704
morie_otis_region_alert_state_richness, 704
morie_otis_regional_placement *(morie_otis_primitives)*, 701
morie_otis_repeat_placement_concentration, 705
morie_otis_rplace, 705
morie_otis_tps_analyze_all, 706
morie_otis_tps_composite_overlay, 707
morie_otis_tps_overlay, 707
morie_otis_tps_per_region_rollup, 708
morie_otis_tps_yoy_correlation, *707*, 708
morie_otis_volat, 709
morie_otis_volatility *(morie_otis_primitives)*, 701
morie_otis_within_year_placement_count, 710
morie_otis_within_year_region_diversity, 710
morie_paired_t_test, 711
morie_parse_nypd_law_code, 711
morie_paths, 712
morie_pca_dimension_reduction, 713
morie_pcg_filter, 713
morie_penalized_regression, 714
morie_percentile_modified_rank, 715
morie_performance_check_collinearity, 716
morie_performance_check_model, 716
morie_performance_check_outliers, 717
morie_performance_r2, 717
morie_point_biserial_r, 718
morie_polynomial_regression, 718
morie_power_prop_test, 719
morie_power_t_test, 720
morie_ppcor_partial, 721
morie_ppcor_semipartial, 721
morie_pps_sample, 722
morie_prediction_accuracy, 723
morie_predpol_aggregate_areas, 723
morie_predpol_calibration_audit, 724
morie_predpol_score_disparity, 725
morie_predpol_temporal_audit, 726
morie_profile_dataset, 727
morie_profile_dataset(), *854*
morie_prophet_components, 727
morie_proportion_ci, 728
morie_psymet_alpha, 729
morie_psymet_alphadel, 729
morie_psymet_ave, 730
morie_psymet_bartlett, 730
morie_psymet_cr, 731
morie_psymet_discrimination, 731
morie_psymet_itemtotal, 732
morie_psymet_kmo, 732
morie_psymet_omega, 733
morie_psymet_parallel, 733
morie_psymet_splithalf, 734
morie_quantile_reg, 734
morie_random_forest_ensemble, 735
morie_random_forest_genomic, 736
morie_random_search_cv, 737
morie_randtests_bartels, 738
morie_randtests_runs, 738
morie_randtests_turning_point, 739
morie_rank_based_test, 739
morie_rank_order_statistics, 740
morie_rank_placements, 741
morie_rdd_bandwidth_cct, 741
morie_rdd_bandwidth_ik, 742

morie_rdd_bandwidth_rot, 742
morie_rdd_bandwidth_sensitivity, 743
morie_rdd_bias_corrected, 743
morie_rdd_cattaneo_density, 744
morie_rdd_covariate_balance, 745
morie_rdd_density_test, 746
morie_rdd_discrete, 746
morie_rdd_donut, 747
morie_rdd_fuzzy, 748
morie_rdd_geographic, 749
morie_rdd_kernel_epanechnikov *(morie_rdd_kernels)*, 750
morie_rdd_kernel_gaussian *(morie_rdd_kernels)*, 750
morie_rdd_kernel_triangular *(morie_rdd_kernels)*, 750
morie_rdd_kernel_uniform *(morie_rdd_kernels)*, 750
morie_rdd_kernels, 750
morie_rdd_kink, 750
morie_rdd_local_polynomial, 751
morie_rdd_local_randinf, 752
morie_rdd_local_randomisation, 753
morie_rdd_mccrary, 753
morie_rdd_placebo_cutoff, 754
morie_rdd_plot_data, 755
morie_rdd_power, 756
morie_rdd_power_calc, 756
morie_rdd_sample_size, 757
morie_rdd_sharp, 758
morie_read_outputs_manifest, 759
morie_recommended_pair_test, 760
morie_recommended_summary, 760
morie_regime_switching, 761
morie_regularization_path, 761
morie_risk_difference_ci, 762
morie_risk_ratio_ci, 763
morie_rkhs_full, 763
morie_rnn_genomic, 764
morie_roc_auc_score, 765
morie_rsample_bootstraps, 765
morie_run_ebac_selection_ipw_analysis, 766
morie_run_morie_module, 767
morie_run_morie_modules, 768
morie_run_pipeline, 769
morie_run_propensity_ipw_analysis, 770
morie_run_treatment_effects_analysis, 771
morie_run_weighted_logistic_analysis, 772
morie_run_workflow_step, 773
morie_sample, 774
morie_sample_size_logistic, 774
morie_semipar_bridge, 775
morie_sensitivity_evalue, 776
morie_sensitivity_konfound, 777
morie_sensitivity_omitted_var_bias, *777*, 778, *1036*
morie_sensitivity_rosenbaum, 779
morie_sensitivity_tipping_point, 780, *1083*
morie_sgolay_smooth, 780
morie_sgolay_smooth(), *1069*
morie_shapiro_wilk_test, 781
morie_sign_test_power, 782
morie_simple_random_sample, 782
morie_simpleboot_two, 783
morie_simulate_longitudinal_panel, 784
morie_siu_all_analyses, 785
morie_siu_analyze, 786
morie_siu_anomaly_check, 786
morie_siu_audit_case, 787
morie_siu_audit_columns, 788
morie_siu_by_police_service, 790
morie_siu_by_year, 791
morie_siu_cache_path, 791
morie_siu_case_counts, 792
morie_siu_charges_by_year_chi2, 792
morie_siu_classify_mandela, *786*, 793
morie_siu_compare, 794
morie_siu_decision_timing, 795
morie_siu_demographics, 796
morie_siu_fetch, 796
morie_siu_fetch_cases, 797, *798*
morie_siu_fetch_dataframe, 798
morie_siu_index, 799
morie_siu_index_url *(morie_siu_fetch)*, 796
morie_siu_llm_extract, 800, *813*
morie_siu_mental_health_race_indicators, 801
morie_siu_parse_html, 802
morie_siu_parse_news_html, 802

morie_siu_parser, 803
morie_siu_record_correction, 803
morie_siu_refresh_manifest, 804
morie_siu_sanity_check, 806
morie_siu_sprott_doob_feb2021, *786*, 807
morie_siu_sprott_doob_iftene_table1, 807
morie_siu_sprott_doob_iftene_table10, 808
morie_siu_sprott_doob_iftene_table15, 808
morie_siu_sprott_doob_iftene_table9, 809
morie_siu_sprott_doob_table11, 809
morie_siu_sprott_doob_table12, 809
morie_siu_sprott_doob_table13, 810
morie_siu_sprott_doob_table15, 810
morie_siu_sprott_doob_table19, 811
morie_siu_sprott_doob_table22, 811
morie_siu_sprott_doob_table23, 812
morie_siu_sprott_doob_table4, 812
morie_siu_translate, 813
morie_siu_translate_fr_to_en *(morie_siu_translate)*, 813
morie_siu_verify_chi2, *792*, 815
morie_siu_verify_published_chi_squares, 815
MORIE_SIUIAP_AFFIDAVITS, 816
morie_siuiap_cite, 816
MORIE_SIUIAP_CRIMSL_REPORTS, 817
MORIE_SIUIAP_ORIGINAL_PANEL_2019_2020, 817
MORIE_SIUIAP_PANEL_MANDATE, 818
MORIE_SIUIAP_PANEL_MEMBERS, 818
morie_siuiap_panel_summary, 818
MORIE_SIUIAP_REPORTS, 819
MORIE_SIUIAP_URL, 819
morie_spatial_voting_aldrich_mckelvey, 820
morie_spatial_voting_alpha_nominate, 821
morie_spatial_voting_anchoring_vignettes, 822
morie_spatial_voting_bayesian_am, 822
morie_spatial_voting_bayesian_irt_likelihood, 823
morie_spatial_voting_bayesian_irt_posterior, 824
morie_spatial_voting_bayesian_mds, 825
morie_spatial_voting_bayesian_unfolding, 826
morie_spatial_voting_blackbox, 826
morie_spatial_voting_cjr_irt, 827
morie_spatial_voting_classical_mds, 828
morie_spatial_voting_cutting_lines, 829
morie_spatial_voting_double_centering, 829
morie_spatial_voting_dw_nominate, 830
morie_spatial_voting_dynamic_irt, 831
morie_spatial_voting_em_irt, 832
morie_spatial_voting_ideal_point_recovery, 832
morie_spatial_voting_indscal, 833
morie_spatial_voting_mds_fit_stats, 834
morie_spatial_voting_mlsmu6, 835
morie_spatial_voting_nominate_bootstrap, 836
morie_spatial_voting_nominate_loglik, 837
morie_spatial_voting_nominate_utility, 838
morie_spatial_voting_nominate_vote_prob, 838
morie_spatial_voting_nonmetric_mds, 839
morie_spatial_voting_nonparametric_bootstrap, 840
morie_spatial_voting_normal_vectors, 841
morie_spatial_voting_optimal_classification, 842
morie_spatial_voting_ordered_oc, 843
morie_spatial_voting_ordinal_irt, 843
morie_spatial_voting_procrustes, 844
morie_spatial_voting_smacof, 845
morie_spatial_voting_smacof_unfolding, 846
morie_spatial_voting_unfolding_stress, 847
morie_spatial_voting_wordfish, 847
morie_spearman_rho, 848

morie_specs_from_df, 849
morie_spectral_cluster, 849
morie_spectral_density, 850
morie_sprott_doob, 851
morie_stat_bridge, 851
morie_stat_commands, 852
morie_state_space_model, 852
morie_stratified_sample, 853
morie_suggest_analysis_plan, 854
morie_sukhatme_test, 854
morie_summarize_output_audit, 855
morie_survey_calibrate, 855
morie_survey_complex_glm, 856
morie_survey_design, 857
morie_survey_glm, 857
morie_survey_hajek_mean, 858
morie_survey_ht_total, 858
morie_survey_mean, 859
morie_survey_poststratify, 859
morie_survey_ratio, 860
morie_survey_subpop, 860
morie_survival_aft, 861
morie_survival_cif, 861
morie_survival_compare_parametric, 862
morie_survival_concordance, 862
morie_survival_cox, 863
morie_survival_coxsnell, 863
morie_survival_deviance, 864
morie_survival_finegray, 864
morie_survival_hr, 865
morie_survival_km, 865
morie_survival_landmark, 866
morie_survival_left_truncated_km, 866
morie_survival_logrank, 867
morie_survival_martingale, 867
morie_survival_nelsonaalen, 868
morie_survival_parametric, 868
morie_survival_rmst, 869
morie_survival_rmst_diff, 869
morie_survival_schoenfeld, 870
morie_survival_turnbull, 870
morie_svm_genomic, 871
morie_svm_hinge_primal, 871
morie_svm_kernel_trick, 872
morie_sync_rng, 873
morie_synth_corrections_uof, 873
morie_synth_otis, 874, *875*
morie_synth_otis_all, 875, *875*
morie_terry_hoeffding_test, 876
morie_tgarch_model, 876
morie_threshold_autoregression, 877
morie_to_hood_crosswalk, 877
morie_to_neighbourhoods, 878
morie_to_neighbourhoods(), *345*
morie_tolerance_limits, 879
morie_toronto_neighbourhoods, 880
morie_tps_add_hood_140_from_158, 880
morie_tps_add_hood_158_from_140, 881
morie_tps_add_hood_158_from_140(), *880*
morie_tps_aggregate_158_to_140, 882
morie_tps_aggregate_158_to_140(), *878*
morie_tps_analyze, 883
morie_tps_analyze_all, *883*, 884
morie_tps_analyze_assault, 884
morie_tps_analyze_autotheft *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_bicycletheft *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_breakandenter *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_communitysafetyindicators *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_csi_from_dataframes, 885
morie_tps_analyze_hatecrimes *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_homicides *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_intimatepartnerandfamilyviolence *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_neighbourhoodcrimerates *(morie_tps_analyze_assault)*, 884
morie_tps_analyze_one, *883*, 886
morie_tps_analyze_robbery *(morie_tps_analyze_assault)*, 884

morie_tps_analyze_shootingandfirearmdiscarges (morie_tps_analyze_assault), 884
morie_tps_analyze_theftfrommovingvehicle (morie_tps_analyze_assault), 884
morie_tps_analyze_theftover (morie_tps_analyze_assault), 884
morie_tps_arima_forecast, 886, *938*
morie_tps_assert_hood_version, 887
morie_tps_available_formats, 887
morie_tps_bivariate_moran, 888, *1087*
morie_tps_bivariate_morans_i, 889
morie_tps_category_correlation_matrix, 890
morie_tps_changepoint_detection, 890, *938*
morie_tps_compare_hawkes_kernels, 891, *904*
morie_tps_composite_index, 892
morie_tps_crime_compare, *883*, *884*, 893
morie_tps_criminal_network_graph, 893, *935*
MORIE_TPS_CSI_CATEGORIES, 894
morie_tps_csi_per_neighbourhood, 895
morie_tps_csi_per_year, 895, *895*
morie_tps_csi_weight, 897
morie_tps_data_dir, 897
morie_tps_dbscan_clusters, 898, *1086*
morie_tps_disaggregate_140_to_158, 899
morie_tps_disaggregate_140_to_158(), *878*, *882*
morie_tps_district_for_centroid, 900, *924*
morie_tps_fetch_category, 900
morie_tps_fetch_category(), *901*
morie_tps_fetch_dataframe, 901
morie_tps_fokker_planck_grid, 902, *937*
morie_tps_getis_ord_g_star, 902, *1086*
morie_tps_gini_concentration, 903
morie_tps_hawkes_advanced, 904
morie_tps_hawkes_advanced_fit, *904*, 905
morie_tps_hawkes_markovian_vs_nonmarkovian, *904*, 906
morie_tps_hawkes_temporal_fit, 907, *937*
morie_tps_inspection_game_phase, 908, *935*
morie_tps_kde_density, 909, *1086*
morie_tps_langevin_simulate, 910, *937*
MORIE_TPS_LAYER_URLS (morie_tps_layer_urls), 910
morie_tps_layer_urls, 910
morie_tps_levy_flight_alpha, 911, *935*
morie_tps_list_categories, 912
morie_tps_list_categories(), *901*
morie_tps_list_datasets, 912
morie_tps_list_formats, 913
morie_tps_load, 913
morie_tps_load(), *914*
morie_tps_load_dataset, 914
morie_tps_load_dataset(), *897*
morie_tps_local_morans_i, 915, *1086*
morie_tps_lotka_volterra_police_crime, 916, *935*
morie_tps_moran_sweep_heatmap, 917, *1087*
morie_tps_morans_i_neighbourhood, 918, *1086*
morie_tps_neighbourhood_concentration, *883*, 919
morie_tps_offence_summary, *883*, 920
morie_tps_polygon_morans_i, *888*, *917*, 920, *1086*
morie_tps_pretty_label, 921, *924*
morie_tps_project_xy, 922, *924*
morie_tps_psdp_layers, 922
morie_tps_psdp_layers(), *345*
MORIE_TPS_REGISTRY, 923
morie_tps_render, 923
morie_tps_render_choropleth, 924, *924*, *929*
morie_tps_render_dbscan, 925
morie_tps_render_district_proportional, 926
morie_tps_render_points, *924*, 926
morie_tps_render_quad, 927
morie_tps_render_satscan_panel, 928
morie_tps_render_yearly_grid, *924*, 928
morie_tps_resolve_hood_col, 929
morie_tps_resolve_hood_col(), *331*
morie_tps_ripley_k, 930, *1086*
morie_tps_sarima_forecast, 931, *937*

morie_tps_sdb_reaction_diffusion, 931, *935*
morie_tps_sdb_turing_demo, 933, *935*
morie_tps_seasonal_pattern, 934, *938*
morie_tps_spatial_summary, *883*, 934
morie_tps_statphysics, 935
morie_tps_statphysics_analyze_all, 936
morie_tps_stochastic, 937
MORIE_TPS_SUPPORTED_FORMATS, *914*, 937
morie_tps_temporal, 938
morie_tps_temporal_summary, *883*, 938
MORIE_TPS_TORONTO_POPULATION_BY_YEAR, 939
MORIE_TPS_TOTAL_CSI_WEIGHTS, 939
morie_tps_urban_scaling_beta, *935*, 940
morie_tps_use_of_force, 941
MORIE_TPS_VIOLENT_CSI_WEIGHTS, 942
morie_tps_year_over_year_trend, *938*, 942
morie_tps_year_to_hood_version, 943
morie_tps_yoy_panel, 943
morie_tpsuof *(morie_tps_use_of_force)*, 941
morie_transformer_genomic, 944
morie_tsne_reduction, 945
morie_two_sample_coverage, 946
morie_two_sample_t_test, 946
morie_unobserved_components, 947
morie_userguide, 948
morie_validate_cpads_data, 948
morie_validate_outputs_manifest, 949
morie_van_der_waerden_test, 949
morie_vecm, 950
morie_verify_statistical_output, 950
morie_vertex_access_token, 951
morie_vertex_ask_gemini, 952
morie_vertex_health_check, 953
morie_vertex_resolve_config, 953
morie_vpd_download_instructions, 953
morie_wavelet_time_series, 954
morie_weights_bootstrap, 955
morie_weights_brr, 955
morie_weights_calibrate_to_totals, 956
morie_weights_combined, 956
morie_weights_deff, 957
morie_weights_design, 957
morie_weights_detect_extreme, 958
morie_weights_diagnostics, 958
morie_weights_ess, 959
morie_weights_fay_brr, 959
morie_weights_greg, 960
morie_weights_jackknife, 960
morie_weights_multiframe, 961
morie_weights_nonresponse, 961
morie_weights_normalize, 962
morie_weights_poststratify, 962
morie_weights_propensity_nonresponse, 963
morie_weights_rake, 963
morie_weights_replicate_variance, 964
morie_weights_sdr, 965
morie_weights_smooth, 965
morie_weights_trim, 966
morie_wilcoxon_power, 966
morie_wilcoxon_signed_rank_test, 967
morie_write_audit_markdown, *42*, 968
morie_write_synthetic_data, 968
morie_xgboost_objective, 969
mrm_anova_bonferroni, 970
mrm_anova_oneway, 971
mrm_anova_power, 972
mrm_arsau, 973
mrm_assumptions_check, 973
mrm_causal_design, 974
mrm_check_balancing, 975
mrm_check_overlap, 976
mrm_classify_mandela, 977
mrm_clt_demo, 979
mrm_design, 980
mrm_diagnostics, 980
mrm_doe, 981
mrm_factorial_2k, 982
mrm_fractional_factorial, 983
mrm_gentrification, 984
mrm_gentrification_panel, 984
mrm_graeco_latin, 986
mrm_kulldorff, 987
mrm_latin_square, 987
mrm_lisa, 988
mrm_mathstats, 989
mrm_mc_power, 989
mrm_median_causal_effect, 990
mrm_morans_i, 991
mrm_oneprop_test, 992

mrm_otis, 993
mrm_otis_mandela_spectrum, 994
mrm_otis_mortification_cooccurrence, 995
mrm_otis_placement_concentration, 996
mrm_otis_region_locality, 997
mrm_otis_seg_duration_km, 998
mrm_perm_block, 999
mrm_pit, 1000
mrm_primitives_ordinal, 1001
mrm_qq_plot, 1002
mrm_random_latin, 1003
mrm_rcbd, 1003
mrm_response_surface, 1004
mrm_samples, 1005
mrm_siu, 1006
mrm_siu_case_to_decision_km, 1006
mrm_siu_outcome_classifier, 1008
mrm_siu_per_service_rate, *786*, 1008
mrm_spatial_spillover, 1009
mrm_spatial_spillover_decomposition, 1010
mrm_standardised_difference, 1011
mrm_synthetic_area_exposure, 1012
mrm_synthetic_exposure, 1013
mrm_threshold_coefficient, 1014
mrm_threshold_specific_ordinal, 1014, *1014*
mrm_tps, 1016
mrm_tps_kulldorff_scan, 1017
mrm_tps_levy_scaling, 1018
mrm_tps_lisa, 1019
mrm_tps_load_hawkes_refit, 1020
mrm_tps_moran_clustering, 1020
mrm_tps_neighbourhood_recurrence_km, 1021
mrm_tps_polygon_moran_per_year, 1022
mrm_two_treatment_test, 1023
mrm_twoprop_test, 1024
mrm_uof, 1025
mrm_uof_data_quality_audit, *123*, *125*, 1025, *1025*
mrm_uof_demographic_disparity, *1025*, 1026
mrm_uof_force_concentration, *123*, *125*, *1025*, 1026
mrm_uof_region_locality, *1025*, 1027
mrm_uof_weapon_diversity, *123*, *125*, *126*, *1025*, 1028
mrm_uof_yoy_change, *123*, *1025*, 1028
mrm_var_test, 1029

n_effective_tests, 1030
n_stat_commands, 1030
nested_cross_validate, 1031
normality_suite, 1032
number_needed_to_harm, 1033
number_needed_to_treat, 1033
nw_regression, *775*, 1034

odds_ratio, 1034
odds_ratio_table, 1035
omega_squared, 1035
omitted_variable_bias, *778*, 1036
one_proportion_ztest, 1037
one_sample_ttest, 1037
one_way_anova, 1038
or_to_d, 1038
or_to_r, 1039

paired_permutation_test, 1039
paired_ttest, 1040
parametric_bootstrap, 1040
partial_correlation, 1041
partial_eta_squared, 1042
pburg, 1042
pcgenv, 1043
pcgmur, 1044
pcgseg, 1044
pearson_correlation, 1045
permutation_fdr, 1046
permutation_fwer, 1046
permutation_test, 1047
pettitt_changepoint, 1048
pfd, 1049
ph_assumption_test, 1050
phi_coefficient, 1050
point_biserial_correlation, 1051
polr, *1001*
prediction_interval, 1051
print.morie_audit_result, 1052
print.morie_tps_result, 1052
print.morie_variable_taxonomy, 1053
probabilistic_bias_analysis, 1053

R6Class, *474*
r_effect_size, 1054

r_squared, 1054
r_to_d, 1055
r_to_or, 1055
ramsey_reset_test, 1056
random_effects_meta, 1056
rank_biserial_correlation, 1057
rate_ratio, 1057
read.csv, *500*, *551*
read_csv, *551*
register_stat_command, *852*, 1058
regression_table, 1058
repeated_cv, 1059
repeated_measures_anova, 1060
resolve_stat_command, *852*, 1060
rgpsd, *465*
risk_difference, 1061
risk_ratio, 1061
rosenbaum_bounds, 1062
rrint, 1062
run_stat_command, *852*, 1063
runs_test, 1064

sampen, 1064
score_data_quality, 1065
score_test, 1065
semi_partial_correlation, 1066
SemiparKernels, *775*, 1066
sensitivity, 1067
sensitivity_rosenbaum, 1067
sensitivity_summary, 1068
sensitivitymv::senmv, *618*
set.seed(), *379*, *380*
sgolay, 1069
sidak, 1070
silverman_bandwidth, *775*, 1070
simes_combined, 1071
spearman_correlation, 1071
specification_curve, 1071
standardized_coefficients, 1072
stat_bridge_exec, 1073
stat_bridge_fn_info, 1073
stat_bridge_fn_search, 1074
stat_bridge_help, 1074
stat_bridge_main, 1075
stat_bridge_registry_json, 1075
stat_bridge_verify, 1076
stat_command, *852*, 1076
statistics, 1077
storey_q, 1078
stouffer_combined, 1078
subsampling, 1079
substance_categories, 1079
summary_statistics_table, 1080

table1, 1081
tables_pub, 1082
tempdir, *147*, *503*, *797*
temporal_validate, 1082
tippett_combined, 1083
tipping_point_analysis, *780*, 1083
tps_crime, 1084
tps_csi, 1085
tps_spatial, 1086, *1086*
tps_spatial_advanced, 1086
treatment_effect_table, 1087
two_proportion_ztest, 1087
two_sample_ttest, 1088
two_way_anova, 1088

utils::install.packages(), *459*, *555*

validate_schema, 1089
validation, 1089
vancouver_crime_adjacent, 1090
vargha_delaney_a, 1091
variable_taxonomy, 1092
variance_equality_suite, 1092
variance_ratio, 1093

wald_test, 1093
welch, 1094
welch_ttest, 1095
wilcoxon_signed_rank, 1095
wild_bootstrap, 1096