Package: creditratingsystem (via r-universe)

October 29, 2026

Type Package

Title Offline Credit Risk Modelling, Calibration and Validation

Version 0.1.1

Description Native credit risk calculations, statistical estimation, calibration, validation and cash flow simulation using local structured data. Methods cover consumer and business lending with explicit modelling assumptions, diagnostics and reproducible reference data.

License AGPL-3

Encoding UTF-8

Depends R (>= 4.1.2)

Imports stats, utils, jsonlite, digest, lpSolve, Matrix, survival, glmnet, MASS, e1071, brglm2, splines, xgboost, copula, mvtnorm, vars, urca, KFAS

Suggests testthat, knitr, rmarkdown

NeedsCompilation no

Author RiskDataScience GmbH [aut, cph], Dimitrios Geromichalos [cre] (Dr.)

Maintainer Dimitrios Geromichalos <riskdatascience@web.de>

Config/pak/sysreqs make libgsl0-dev

Repository https://rds0001.r-universe.dev

Date/Publication 2026-10-29 16:17:16 UTC

RemoteUrl https://github.com/cran/creditratingsystem

RemoteRef HEAD

RemoteSha f685e5a40e6be51b37a72163bd203c0a26af0125

Contents

affordability 4
affordability_stress_backtest 7

application_risk . . . 9
assess_counterfactual_robustness . . . 12
asset_pool_backtest . . . 13
behavioral . . . 15
build_calculation_context . . . 20
calibrate_intercept . . . 21
calibration_models . . . 24
classification_bayesian . . . 31
classification_flexible . . . 33
classification_models . . . 36
classification_neural . . . 40
collections . . . 42
collections_backtest . . . 47
compile_credit_decision_policy . . . 49
compute_credit_ale . . . 51
compute_credit_ice . . . 52
content_fingerprint . . . 54
counterparty_exposure_backtest . . . 55
covered_bond_dual_recourse . . . 56
credit_data . . . 58
credit_dependence . . . 61
credit_links . . . 64
credit_loss_distribution . . . 68
credit_permutation_importance . . . 70
cumulative_pd_from_conditional . . . 71
decision_policy_backtest . . . 72
decisioning_offer_economics . . . 74
decisioning_policy_evaluation . . . 76
detect_cashflow_default . . . 79
ead . . . 80
ead_counterparty . . . 82
ead_models . . . 85
estimate_long_run_pd . . . 87
estimate_simulation_risk . . . 89
estimate_tranche_credit_metrics . . . 90
expected_loss . . . 92
expert_probability . . . 93
expert_qualitative . . . 97
explainability_attribution . . . 99
features . . . 102
find_feasible_credit_counterfactual . . . 104
fit_binary_glm . . . 106
hybrid . . . 109
importance_sample_defaults . . . 112
lgd . . . 113
lgd_downturn . . . 115
lgd_models . . . 116
lgd_workout . . . 119

loss_paths . . . 122
macro . . . 126
market_implied . . . 133
monitoring . . . 145
monitoring_metrics . . . 146
pd_term_structure_extended . . . 151
pit_ttc_transform . . . 153
portfolio_simulation . . . 155
predict_cure_and_redefault . . . 158
rank_credit_reason_codes . . . 160
rating_system . . . 161
reconcile_card_authorizations . . . 166
retail_behavior . . . 168
retail_cashflows . . . 170
retail_evidence . . . 171
retail_features . . . 174
retail_obligations . . . 176
retail_pool_diagnostics . . . 178
retail_pools . . . 180
retail_sme . . . 183
run_tranche_waterfall . . . 185
scorecard . . . 186
selection_bias . . . 189
selection_bias_sensitivity . . . 194
serialize_fitted_model . . . 196
simulate_asset_pool . . . 197
simulate_contract_counterparties . . . 198
simulate_recovery_waterfall . . . 200
simulate_trade_receivables . . . 201
simulate_transactions . . . 202
simulation_cashflows . . . 204
simulation_drivers . . . 206
stress_pool_counterparties . . . 207
structural . . . 209
structured_credit_pool_foundations . . . 213
survival_models . . . 215
synthetic . . . 221
transfer_events . . . 224
validate_probability . . . 226
validate_rating_scale . . . 227
validate_source_lineage . . . 228
validate_system_coverage . . . 230
validation . . . 232
validation_backtests . . . 239
validation_benchmarks . . . 243
validation_interventions . . . 245
validation_numerics . . . 247
validation_retail . . . 249

vintage_backtest 251

Index **253**

`affordability` *Explicit Affordability Ratios and Income Surplus*

Description

Calculate stock and service ratios and signed income surplus, then apply user-provided thresholds independently of PD and credit approval.

Usage

```
debt_to_income(debt_stock, income, income_frequency, currency,
    ratio_period,
                income_basis = "net", context = NULL)
debt_service_to_income(debt_service, income, period, currency,
                       income_basis = "net", context = NULL)
residual_income(net_income, recurring_expenses,
    existing_debt_service,
                 new_debt_service, period, currency, context = NULL)
assess_affordability(metrics, threshold_profile, evidence,
    context = NULL)
normalize_income_period(components, target_period, target_basis,
    currency,
  payments_per_year, variable_income_factor, context = NULL)
estimate_recurring_expenses(expenses, target_period, currency,
  payments_per_year, household_profile, benchmarks = NULL,
    context = NULL)
combined_loan_to_value(claims, collateral, allocations, currency,
    context = NULL)
stress_affordability(baseline, scenario, context = NULL)
```

Arguments

`debt_stock`

Nonnegative debt stock.

`income`, `net_income`

Finite period income; negative income is allowed.

`income_frequency`, `period`

One of weekly, biweekly, four_weekly, monthly, quarterly or annual.

`currency`

Explicit common currency.

`ratio_period`

Annualize income with `"annual"`, or retain `"income_period"`.

`income_basis`

Explicit `"net"` or `"gross"` basis.

`debt_service`, `existing_debt_service`, `new_debt_service`

Nonnegative, same-period service amounts.

`recurring_expenses`

Nonnegative expenses, disjoint from debt service.

`metrics`

Named list of finite scalar metrics or NULL for unavailable metrics.

`threshold_profile`

Named list with schema_version 1.0, policy_id, version, required_evidence and rules. Each rule has rule_id, metric, operator and threshold.

`evidence`

Named list of logical evidence-availability flags.

`components`

Unnamed list of income records: economic_id, amount, frequency, basis, currency, kind (recurring, variable, internal_transfer). Crossing net/gross basis requires same-period tax_amount.

`target_period`, `target_basis`

Target frequency key and net/gross income basis.

`payments_per_year`

Named list of explicit positive annual payment counts, including the target and every source frequency.

`variable_income_factor`

Inclusion fraction in [0,1] for positive variable earnings. Negative earnings are fully retained.

`expenses`

Unnamed list of records: economic_id, category, amount, frequency, currency, kind (essential, discretionary, debt_service, internal_transfer). Null amounts denote unknown expenditure.

`household_profile`

Named list with profile_id and array required_categories.

`benchmarks`

Optional local rows with profile_id, category, amount, frequency, currency, kind. Matching category amounts are floors.

`claims`

Secured claim records with claim_id, amount and currency.

`collateral`

Collateral records with collateral_id, value, currency, valuation_date and declared freshness CURRENT or STALE.

`allocations`

Rows with claim_id, collateral_id, amount and integer priority. Amounts must partition each claim once across assets.

`baseline`

Named household object with as_of, reporting currency, liquid_buffer, incomes, expenses, contracts, collateral, cltv_scope and thresholds. See details for the monthly contract schema.

`scenario`

Named object with schema_version 1.0, scenario_id and monthly periods. Each period supplies date, income_factors, expense_factors, collateral_factors, rate_shifts and fx_to_reporting.

`context`

Optional calculation context or its result.

Details

Length-one amounts expand; other vectors require equal lengths. Annualization uses 52 weekly, 26 biweekly, 13 four-weekly, 12 monthly, 4 quarterly and 1 annual payment. Nonpositive income returns a NULL ratio and NONPOSITIVE_INCOME status. No universal thresholds or approval rules are supplied. Threshold comparisons are unrounded, and strict inequalities fail at equality. Missing inputs give INSUFFICIENT_EVIDENCE; known breaches remain visible. Otherwise the assessment is AFFORDABLE or NOT_AFFORDABLE. Household obligations must be reconciled before supplying disjoint totals.

Income normalization instead uses the explicitly supplied payment calendar. Duplicate economic sources count once; conflicting economics fail. Transfers contribute zero. Unknown included income or tax bridges make total income NULL. No taxes are estimated. Expense reconciliation takes the maximum known normalized amount per economic obligation, then applies category floors. It does not infer recurring obligations from raw transactions. Debt service is separate from living costs; unknown debt service stays NULL. Missing required categories and obligations remain explicit.

CLTV counts each allocated collateral once. Lower ranks consume available value first and equal ranks share proportionally. Claim allocation reconciliation uses relative or absolute tolerance 1e-12. Zero collateral value gives NULL ratios. Rank coverage is not an LGD estimate or a legal enforceability assessment.

Stress baseline incomes have economic_id, monthly_amount, net basis and currency; expenses have economic_id, monthly_amount and currency. Contracts have economic_id, currency, principal, remaining_months, annual_rate, repayment (annuity or interest_only), balloon_amount, rate_type (fixed or reset), reset_dates, rate_floor and rate_cap. Rates are nonnegative nominal annual rates divided by twelve. Interest-only requires balloon_amount equal to principal. Stress collateral records

use economic_id, value, currency. cltv_scope must declare all_contracts_secured_by_pool or not_applicable. Thresholds explicitly contain minimum_residual, maximum_dsti and maximum_cltv.

Scenario dates are consecutive monthly anniversaries clamped to month-end. Every factor map covers the respective baseline economic IDs exactly, including empty named maps for absent groups. FX is reporting-currency units per source unit, with reporting FX equal to one. Rate shifts apply relative to original coupon only at declared resets, bounded by floor and cap. Projected resets must coincide with payment dates. Annuity payment solves the present-value equation including the terminal balloon and is recomputed at reset. At maturity, actual remaining principal plus interest is due; no refinancing is assumed. Fixed coupons ignore shifts. Residuals accumulate into liquidity, with first strictly negative buffer date reported. Projection continues after exhaustion to expose the funding gap; no default or collections transition is inferred. Annual DTI uses opening debt and current monthly income times twelve; CLTV uses closing debt and the declared common collateral pool. No independent shocks are drawn. Arrays are unnamed lists; objects are named lists, including empty maps.

Value

A CalcResult with ratios and denominator status, signed residuals and component trace, or a separate affordability assessment with rules and headroom.

Examples

```
debt_to_income(60000, 2500, "monthly", "EUR", "annual")$value
debt_service_to_income(900, 3000, "monthly", "EUR")$value
residual_income(3000, 1000, 900, 200, "monthly", "EUR")$value
```

[1.5in]

`affordability_stress_backtest`
Verify Frozen Household Stress Assessments

Description

Checks historically verified economic evidence, household source allocations and frozen contractual stress projections, with separately observed factual payments.

Usage

```
affordability_stress_backtest(cases, economic_evidence, allocations,
  design, payment_outcomes = NULL, tolerance = 1e-10,
  uncertainty = "none", cluster_resamples = NULL,
  confidence_level = .95, context = NULL)
```

Arguments

`cases`

Common dated validation cases, household IDs, genuine baseline/scenario/assessment payloads, dated threshold profiles, expected economic IDs and exact evidence bindings.

`economic_evidence`

Dated verified economic source records with normalized monthly net income or explicit gross-to-net reconciliation and complete contractual terms.

`allocations`

Dated household proportional shares or explicit full-recourse contractual liability assignments.

`design`

Frozen model/data/split/population, currency, target horizon and evaluation cutoff.

`payment_outcomes`

Optional dated complete-capture factual scheduled and paid amounts.

`tolerance`

Nonnegative numerical reconciliation tolerance.

`uncertainty`

`none` or supplied `cluster_bootstrap`.

`cluster_resamples`

Shared whole-case-cluster draws for factual payment metrics.

`confidence_level`

Interior percentile coverage level.

`context`

Optional context consistent with the design.

Details

The actual stress engine is rerun with frozen inputs and compared with the original assessment. Unknown or late income is incomplete evidence, not an invented zero. Proportional household shares cannot duplicate an economic source; full recourse is explicit and contractual only. Gross less deductions reconciles to normalized monthly net income. Contractual resets, amortization, FX and policy thresholds remain governed by the original stress input.

Observed scheduled-minus-paid metrics are descriptive facts under complete mature capture; they are not presumed realizations of the scenario debt service. Positive residual income is not zero PD or a lending approval.

Value

A CalcResult with a ValidationResult, verified projection flags, reconciliation findings, separate factual payment metrics, uncertainty and missing/censored period coverage.

[1.5in]

application_risk
Application evidence, selection-aware targets and conditional risk

Description

Freeze dated applicant evidence, retain joint legal roles, construct funded first-default targets and apply explicit conditional models locally.

Usage

```
build_application_snapshot(applications, feature_records,
    term_events = NULL,
                           required_features = NULL)
assemble_joint_application(application_id, parties, income_items,
                           obligation_items, currency)
build_application_targets(applications, decisions, bookings,
    default_events,
                          exits, horizon_days, evaluation_cutoff,
                          performance_origin = "first_draw_at")
predict_application_risk(snapshot, model, target, calibrator = NULL,
    offset = NULL)
compare_offer_risk(snapshot, offers, model, target, support_bounds,
                   calibrator = NULL, lgd = NULL, offset = NULL)
```

Arguments

applications

Rows with unique application_id, customer_id, product_id, population_id, application_at, information_cutoff, and requested_terms named list.

feature_records

Rows with application_id, feature, value (possibly NULL), observed_at, known_at and source_id.

term_events

Optional rows with application_id, stage (offered/accepted/booked), terms named list, effective_at and known_at.

required_features

Optional names whose missing evidence is reported.

application_id

String application identifier.

`parties`

Rows with party_id, role (borrower/co_borrower/guarantor), liability_share in [0,1], optional household_id.

`income_items`, `obligation_items`

Rows with economic_id, party_id, amount_monthly and currency. Amount is the total shared economic amount.

`currency`

Explicit common currency for supplied monthly amounts.

`decisions`

Rows with decision_id, application_id, status (approved/rejected/pending/withdrawn), decided_at, known_at; optional accepted_at requires accepted_known_at.

`bookings`

Rows with unique application_id and facility_id, booked_at, known_at; optional first_draw_at requires first_draw_known_at. Optional followup_end and outcome_available_at govern follow-up completeness.

`default_events`

Rows with event_id, facility_id, default_at, known_at.

`exits`

Rows with facility_id, exit_at, known_at; early repayment is censoring.

`horizon_days`

Positive integer calendar-day outcome horizon.

`evaluation_cutoff`

ISO date of known events and outcomes.

`performance_origin`

Explicit first_draw_at or booked_at target origin.

`snapshot`

One frozen signed snapshot from build_application_snapshot.

`model`

Frozen ModelArtifact or FitResult with application_contract metadata.

`target`

Complete fitted target, including performance_origin; physical PD.

`calibrator`

Optional calibration artifact matching target and model version.

`offset`

Optional scalar link-scale offset matching fitted classifier.

`offers`

Rows with unique offer_id and terms named list.

`support_bounds`

Named list mapping every modelled term to [lower,upper].

`lgd`

Optional fixed LGD in [0,1], used with offered amount as fixed EAD.

Details

Snapshots select feature revisions with observed_at and known_at no later than information_ cutoff, then latest observation and knowledge dates. Ties fail. Source timestamps and requested/offered/accepted/booked terms stay separate. Snapshots have a shared canonical hash; subsequent inference rejects changed content. Objects remain mutable in memory but are copied and integrity checked.

Joint items are deduplicated within income and debt tables by economic_id. Conflicting shared amounts fail. Guarantor resources do not become borrower income. Original liability roles and shares remain; shares need not sum to one. Budget amounts are not multiplied by liability shares. No individual PDs are averaged, and no household identity is inferred. Caller supplies already normalized, recognized monthly amounts in a common currency.

Targets preserve rejected, approved-not-taken-up, accepted-not-booked, booked-not-drawn and funded stages. Known bookings require approval and acceptance; chronology and separate knowledge dates are checked. Information cutoff cannot follow evaluation cutoff. Acceptance attached to a nonapproved decision fails. A target is first default in (origin, origin+horizon], using funded-facility events. Unfunded applications, early exits, incomplete follow-up and pending ascertainment have missing labels. A complete nondefault window yields zero; default at horizon end yields one. Information cutoff remains separate from performance origin. Calendar maturity is reported separately from label_observed. Booking-origin targets explicitly treat booked facilities as the funded risk set even before first draw. One booked facility per application is supported; split multi-facility records explicitly upstream.

Prediction requires model$application_contract with product_ids, population_ids, performance_ origin, term_stage and term_features. The last maps term field to fitted feature name and may be an explicitly named empty list. Term-feature names must not also occur among customer features. Product, population and complete target must match; missing/nonfinite model inputs fail. The shared classifier currently supports binary GLM artifacts with identity preprocessing. Optional calibration validates target/version; isotonic additionally declares metadata$input_scale = "probability". Output contains PD, unadjusted PD, model and snapshot identities, target and feature coverage. Risk is separate from creditworthiness budgets and approval decisions.

Offer comparisons require an offered-term conditional model and every scenario term to appear in its feature map. All customer features stay fixed. Marginal support bounds flag extrapolation but cannot establish joint support or causal effects. Optional expected loss is PD times fixed LGD times offered amount; requested_terms$currency is then mandatory and returned. No dynamic EAD, take-up, automatic policy or causal claim is implied.

Value

CalcResult. Snapshots, targets and offer results return lists of row lists; joint application and risk prediction return named records. IDs are strings, dates ISO YYYY-MM-DD. Missing values remain NULL.

Examples

```
a <- list(list(application_id = "A", customer_id = "C", product_id = "consumer",
               population_id = "new", application_at = "2024-01-01",
              information_cutoff = "2024-01-01", requested_terms = list(amount = 1000)))
f <- list(list(application_id = "A", feature = "income", value = 3000,
             observed_at = "2023-12-01", known_at = "2023-12-02", source_id = "payroll"))
build_application_snapshot(a, f, required_features = "income")$value
```

[1.5in]

`assess_counterfactual_robustness`
Shared Finite Counterfactual Robustness Scenarios

Description

Evaluate candidate feasibility and target success across declared model and measurement scenarios, retaining every failed variant in the denominator.

Usage

```
assess_counterfactual_robustness(candidates, record, model_variants,
  scenarios, feasibility_contract, target, candidate_ids,
    nominal_variant_id,
  context = NULL)
```

Arguments

`candidates`

Finite candidate matrix or row lists in frozen feature order.

`record`

Numeric baseline vector.

`model_variants`

Named list of predictor and model_contract entries sharing feature order, scale and target. Policy targets also require policy_evaluator and policy_contract.

`scenarios`

Distinct scenario_id, model_variant_id, additive delta vector and nonnegative weight records. At least one weight is positive.

`feasibility_contract`

Declared structural feasibility contract as for ICE.

`target`

Versioned MODEL_OUTPUT boundary/interval or POLICY_ACTION target as for finite counterfactual search.

`candidate_ids`

Distinct candidate identifiers.

`nominal_variant_id`

Named variant used for the separately reported unperturbed nominal evaluation.

`context`

Optional common calculation context.

Details

The same additive measurement perturbation applies to baseline and all candidates. Immutable features mean no action, not no measurement uncertainty. Constraints are checked again. Every supplied scenario remains in the normalized denominator, including failed evaluations. A batch callback failure makes that entire scenario unevaluable; no selected rows are silently retried. Feasibility bounds assign unknown feasibility to failure and success respectively. Worst/best target margins are NULL if any margin is unavailable; policy targets have no numeric margin. All-scenario success includes zero-weight scenarios, while weighted fractions follow declared mass. Feature ranges and failure co-occurrence are descriptive, not causal attribution. Finite scenario fractions are not real intervention probabilities or continuous worst-case guarantees.

Value

CalcResult containing nominal and scenario candidate results, conservative success fractions, feasibility bounds, missing-evaluation mass, margins, perturbation summaries and contract hashes.

See Also

`compute_credit_ice`, `find_feasible_credit_counterfactual`

[1.5in]

`asset_pool_backtest`
Observed Asset-Pool and Tranche Cash Validation

Description

Replay frozen pool forecasts and reconcile independent actual servicing, dated ownership, pool accounts and tranche cash.

Usage

```
asset_pool_backtest(asset_pool, pool_inputs, forecast_paths,
    forecast_inputs,
  servicing, allocations, pool_accounts, coverage, design,
    cash_contract,
  tranche_contracts = list(), tranche_payments = list(),
  ownership_history = NULL, tolerance = 1e-10, context = NULL)
```

Arguments

`asset_pool`

Frozen CreditAssetPool artifact.

`pool_inputs`

Original pool builder keyword arguments.

`forecast_paths`

Frozen AssetPoolPaths artifact.

`forecast_inputs`

Original simulator keyword arguments.

`servicing`

Independent dated whole-asset principal and recovery ledger.

`allocations`

Event-to-pool ownership shares and complete allocated amount maps.

`pool_accounts`

Dated actual opening/closing cash and all receipts/disbursements.

`coverage`

Per-asset completeness, through date, known-at, currency and exact target.

`design`

Common physical validation design, currency and origin-known forecast date.

`cash_contract`

Origin-known opening cash, date and currency.

`tranche_contracts`

Origin-known tranche identifiers, notionals and currency.

`tranche_payments`

Dated actual tranche principal and interest.

`ownership_history`

Optional dated ownership assignments, otherwise original assignments.

`tolerance`

Nonnegative reconciliation tolerance.

`context`

Compatible optional calculation context.

Details

Whole horizons mature before default/prepayment metrics. Incomplete coverage is unknown. Recovery ownership follows the referenced default claim; post-default claim sales require a separate ledger. Original static holdings must match for forecast comparisons. Dynamic or changed holdings receive ledger audits without risk-set-mismatched forecast scores. Tranche cash and principal caps are checked, but contractual waterfall priority, coupons and deficiency ledgers are not inferred. Purchase/sale consideration and completeness are supplied evidence. Forecast scenario weights do not yield empirical confidence intervals. A clean ledger is not release approval.

Value

CalcResult with ValidationResult metrics, explicit applicability, coverage exclusions, ledger violations and cash residuals.

References

BCBS CRE40.93, dated 2019-12-15, transparency of receipts, payments and ledger entries. https://www.bis.org/committees/bcbs/basel-framework/standard/cre/40/inforce/2022-01-01/published/2019-12-15

[1.5in]

`behavioral`
Dated Behavioral Account Risk and State Dynamics

Description

Reconstruct payment arrears, build dated account snapshots and first-event targets, estimate empirical transitions and calculate descriptive vintage performance. Active zero-balance accounts remain in their risk set.

Usage

```
compute_days_past_due(obligations, payments, as_of, known_at,
  allocation_policy = "fifo", currency, context = NULL)
assign_delinquency_state(dpd, past_due_amount, exposure, events,
  state_profile, context = NULL)
build_account_month_panel(accounts, statements, account_events,
```

```
  observation_dates, reporting_lag_days = 0, obligations = NULL,
  payments = NULL, context = NULL)
build_behavioral_targets(panel, events, horizon_months,
    target_event,
  competing_events, exit_treatment, evaluation_cutoff, coverage,
  context = NULL)
estimate_retail_roll_rates(panel, states, from_date, to_date,
    group_by,
  weighting, missing_successor, context = NULL)
project_retail_roll_rates(initial_distribution, transition_matrices,
  states, default_states, closure_states, exposure_by_period,
  context = NULL)
predict_behavioral_risk(snapshot, model, target, calibrator = NULL,
  offset = NULL, context = NULL)
build_vintage_performance(accounts, panel, events, months_on_book,
  evaluation_cutoff, coverage, default_event, delinquency_states,
  exit_events, context = NULL)
```

Arguments

`obligations`

Rows with unique obligation_id, account_id, due_at, known_at, nonnegative amount and currency.

`payments`

Rows with unique payment_id, account_id, paid_at, known_at, amount, currency, kind (payment/reversal). Reversals require reverses_payment_id.

`as_of`, `known_at`

ISO observation and information-cutoff dates.

`allocation_policy`

fifo applies net available cash to oldest due obligations; explicit restricts each payment to its obligation_id.

`currency`

Common currency of the single-account payment calculation.

`dpd`, `past_due_amount`, `exposure`

Nonnegative integer arrears days and finite amounts.

`events`

Dated event rows for target/vintage methods; named boolean event flags for state assignment. Event rows require event_id, account_id, event_type, occurred_at and known_at. Optional LOSS/RECOVERY rows require amount and currency.

`state_profile`

Named profile with schema_version, profile_id, immaterial_state, materiality, buckets, event_priority, optional dpd_default_rule.

`accounts`

Rows with unique account_id, customer_id, product_id, opened_at, known_at, currency. Vintage additionally requires original_balance.

`statements`

Rows with statement_id, account_id, observed_at, known_at, balance, credit_limit, dpd, past_due_amount and currency.

`account_events`

Dated lifecycle events: ACTIVE, DEFAULT, CHARGEOFF, CURED, CLOSED, PREPAID or SOLD.

`observation_dates`

Distinct calendar-month-end ISO dates.

`reporting_lag_days`

Nonnegative conservative evidence delay: subtract from observation date to obtain the information cutoff.

`panel`

Account state rows. Targets require account_id, customer_id, as_of, at_risk, ever_default; transition/vintage methods require state.

`horizon_months`

Positive integer monthly-anniversary target horizon.

`target_event`, `default_event`

Explicit first-event/default event name.

`competing_events`, `exit_events`

Distinct competing exit event names.

`exit_treatment`

competing assigns zero to a first exit; censor leaves its target unobserved. Neither estimates hypothetical continuing-account risk.

`evaluation_cutoff`

Date at which outcome evidence is evaluated.

`coverage`

Rows with account_id, observed_through and known_at, explicitly declaring outcome observation completeness.

`states`

Ordered distinct state names.

`from_date`, `to_date`

Increasing ISO transition endpoint dates.

`group_by`

Origin grouping column names, possibly empty.

`weighting`

count or opening_balance; latter needs balance or features.balance.

`missing_successor`

error or censor; censor reports missing mass separately and estimates transitions conditional on observed successors.

`initial_distribution`

Nonnegative state probability vector summing to one.

`transition_matrices`

One row-stochastic numeric matrix or nested row list per period; unresolved null rows cannot be projected.

`default_states`, `closure_states`

Disjoint individually absorbing states.

`exposure_by_period`

Nonnegative period-by-state matrix of separately supplied conditional exposures. Closed-state exposure must be zero.

`snapshot`

A fingerprinted account snapshot from the panel builder.

`model`

Frozen model artifact or FitResult with behavioral_contract.

`target`

Exact physical event, horizon, time-unit and conditioning contract.

`calibrator`

Optional frozen calibrator compatible with target/model version.

`offset`

Optional offset required by the fitted model.

`months_on_book`

Distinct nonnegative calendar month ages. MOB zero ends at the origination month's final day.

`delinquency_states`

States counted in the current delinquency stock.

`context`

Optional calculation context.

Details

Payment reversals are dated and never rewrite an earlier information cutoff. DPD excludes grace/materiality rules. Amounts due today have zero days past due and are excluded from past_due_amount, but retained as unpaid_due_amount. FIFO is a current net-cash allocation contract, not a bank servicing waterfall.

State materiality contains absolute, relative thresholds and and/or logic. Positive arrears and threshold equality qualify. At zero exposure positive arrears satisfy relative materiality. Buckets have state, lower inclusive and upper exclusive integer bounds, gap-free from zero to a final NULL upper bound. event_priority records have event, state, at_risk and absorbing. Explicit events precede the optional dpd_default_rule (minimum_dpd, state, material_required). No bucket is a universal regulatory default definition.

Panel snapshots choose the latest economic statement/revision available by the cutoff. Statements may use a different billing-cycle date. Missing statements produce missing features, not zero balances. Account features include balance, credit_limit, dpd, past_due_amount and utilization. Legal termination is retained; default history survives cure. Customer sums are currency-specific and not PDs.

Targets use (as_of,horizon_end] first-event windows. Previously defaulted and terminated accounts are outside that risk set. Observed events can identify positive labels early. Non-events require explicit mature coverage. Unknown follow-up is NULL when neither coverage nor event is available. Same-day competing target/exit events require upstream resolution. Overlapping account windows and shared customers are dependent observations.

Roll-rate outputs reconcile origin counts/masses to observed destinations plus censoring. Empty observed rows are NULL, never identity rows. Opening-balance weighting measures monetary flows, not account probabilities. Projection keeps first-default states absorbing and separates transition probabilities from conditional exposures. balance_flows are probability flows times destination exposure, not cash transfers. Supplied exposures must reflect drawdowns, amortization and maturity independently of the probability matrix.

Vintage cells use fixed origination-account denominators. Immature or incompletely observed cells remain NULL. Current delinquency stock differs from ever-default; losses are supplied observed loss/recovery cashflows, not ultimate estimates. Age plus cohort equals calendar period, preventing unrestricted causal separation.

Prediction validates snapshot hash, first-default account eligibility, product/state support and maximum_source_age_days from behavioral_contract. The contract requires risk_unit account and event_policy first_default. It invokes supported frozen shared model prediction, then optional calibration. Model-training and calibration knowledge dates cannot exceed the snapshot cutoff. An active zero balance does not force PD to zero. Isotonic calibration must declare metadata input_scale probability. Model actions and future EAD remain separate. Arrays use unnamed lists and objects named lists.

Value

A CalcResult containing reconciled payment records, state assignments, account/customer panels, event labels, empirical transition tables, projected probability/balance paths, behavioral risk predictions or vintage cells.

[1.5in]

`build_calculation_context`
Build an Explicit Reproducible Calculation Context

Description

Records explicit time, model and data identities with a stable SHA-256 context hash. The system clock is never consulted.

Usage

```
build_calculation_context(as_of_date = NULL, knowledge_date = NULL,
  horizon = NULL, time_unit = NULL, currency = NULL,
  probability_measure = NULL, conditioning = NULL,
  model_version = NULL, data_manifest = NULL, scenario_id = NULL,
  policy_id = NULL)
```

Arguments

`as_of_date`

Observation date as a real ISO YYYY-MM-DD calendar date.

`knowledge_date`

Knowledge date with the same strict representation. Both dates are supplied together or omitted together.

`horizon`

Optional positive finite scalar horizon, paired with a time unit.

`time_unit`

`"days"`, `"months"` or `"years"`.

`currency`

Optional nonempty unit identifier; no exchange conversion is performed.

`probability_measure`

Optional `"PHYSICAL"` or `"RISK_NEUTRAL"`.

`conditioning`

Optional nonempty description of probability conditioning.

`model_version`

Optional model version identifier.

`data_manifest`

Optional named list of JSON-compatible data identity information.

`scenario_id`

Optional scenario identifier.

`policy_id`

Optional local policy identifier.

Details

Without dates the mode is TIME_INDEPENDENT. With both dates the mode is DATED; the dates can be in either order to represent forecast or replay contexts. Hashing sorts object keys and uses a specified binary64 numeric representation, preserving text exactly. Named lists are JSON objects; unnamed lists are arrays. Length-one atomic values are scalars. An unnamed list expresses an explicit empty or length-one array, while `structure(list(), names = character())` expresses an empty object. Object keys must be unique printable ASCII text; values may contain UTF-8 text. Non-finite values are rejected. For cross-language interchange, avoid integral JSON literals outside the exact binary64 integer range; represent large non-integer quantities using floating-point notation.

Value

A `CalcResult` whose `value` is a context named list with a `context_hash`. Context-aware functions accept this value or the whole result.

Examples

```
ctx <- build_calculation_context("2025-12-31", "2026-01-15",
  horizon = 1, time_unit = "years", currency = "EUR",
  probability_measure = "PHYSICAL", conditioning = "survival")
expected_loss(0.02, 0.4, 100000, context = ctx)$unit
```

[1.5in]

calibrate_intercept
Fit and Apply Frozen Probability Calibration

Description

Fit an intercept shift, logistic recalibration, or weighted isotonic step map with explicit target and sample metadata. Apply the frozen artifact with target and model-version checks. All calculations are native R and offline.

Usage

```
calibrate_intercept(base_pd, target_default_rate, weights = NULL,
  metadata, context = NULL, tolerance = 1e-12, max_iterations = 200)
calibrate_logistic(base_pd, outcomes, weights = NULL, metadata,
  context = NULL, tolerance = 1e-10, max_iterations = 100)
calibrate_isotonic(scores, outcomes, weights = NULL, metadata,
  increasing = TRUE, out_of_bounds = "error", context = NULL)
apply_calibrator(calibrator, values, target, base_model_version,
  context = NULL, groups = NULL)
```

Arguments

`base_pd`

Numeric probability vector. Closed unit interval for intercept shifts; strictly interior probabilities for logistic calibration.

`target_default_rate`

Scalar weighted target default rate.

`weights`

Nonnegative integer frequency weights of matching length with positive finite sum. Defaults to one per row. Zero-weight rows do not identify parameters or define isotonic support.

`metadata`

Named list containing `target`, `calibration`, `base_model_version`, `population_definition`, `outcome_policy`, `weight_type="frequency"` and `data_origin`. See Details.

`groups`

Optional aligned group IDs for a hierarchical calibration artifact; see calibration_models.

`context`

Optional calculation context. Any supplied horizon, unit, probability measure and conditioning must agree with the target.

`tolerance`

Positive numerical convergence tolerance smaller than 0.01.

`max_iterations`

Positive integer iteration limit.

`outcomes`

Fully observed numeric binary first-default outcomes.

`scores`

Finite numeric scores, not necessarily probabilities.

`increasing`

Whether calibrated probability is nondecreasing in score. Set to `FALSE` for a nonincreasing map.

`out_of_bounds`

Frozen isotonic support policy: `"error"` rejects scores outside positive-weight calibration support; `"clip"` uses the nearest endpoint probability and counts affected inputs.

`calibrator`

Calibration `CalcResult` or its plain artifact list.

`values`

Numeric vector to transform, in its original order. Empty application vectors are allowed.

`target`

Explicit target list matching the frozen calibration target.

`base_model_version`

Version matching the calibrated base model.

Details

The target contains `event`, positive `horizon`, `time_unit` (days, months or years), `probability_measure` (PHYSICAL or RISK_NEUTRAL) and `conditioning`. Calibration metadata contains ISO dates `start`, `end`, `known_at` with start no later than end no later than known-at, a nonempty `split_id`, and `prediction_origin`. Empirical fitting requires HOLDOUT or OUT_OF_FOLD; EXTERNAL_TARGET is additionally allowed for intercept shifts. Data origin is USER_SUPPLIED, SYNTHETIC or REFERENCE. These are caller declarations: the functions do not independently establish sample independence or outcome maturity.

Intercept calibration solves the weighted mean equation for $\mathrm{logit}^{-1}(\mathrm{logit}(p) + a)$ while retaining structural zero/one probabilities. Finite-shift endpoint targets and a sample without interior positive-weight probabilities are rejected.

Logistic calibration fits unpenalized binomial maximum likelihood on the base logit. Constant logits, complete separation and quasi separation are reported as controlled failures. Endpoints are never silently clipped. Nonpositive slope is explicitly diagnosed. Fisher covariance assumes independent frequency repetitions; it is not cluster-robust uncertainty.

Isotonic calibration first combines tied scores, then uses weighted pool-adjacent-violators. Its application is left-constant on intervals $[x_i, x_{i+1})$; there is no linear interpolation. Original input order is preserved. The support policy is stored in the artifact.

Artifacts carry parameters, target, metadata, diagnostics, an input fingerprint and a checked content hash. To exchange across languages, place the artifact in an additional `calibration_artifact` field of a valid base model and use `serialize_fitted_model`/`load_fitted_model`. These retain full numeric precision. Rounded JSON may fail integrity checks intentionally. Hashes detect changed content; they do not authenticate authorship.

Value

A `CalcResult`. Fit functions return a versioned CalibrationArtifact in `$value`; application returns a probability vector. Diagnostic and provenance fields remain separate. No result implies validation or approval.

References

https://scikit-learn.org/stable/modules/calibration.html

https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.isotonic_regression.html

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html

Examples

```
metadata <- list(
  target = list(event = "first_default", horizon = 12, time_unit = "months",
                probability_measure = "PHYSICAL", conditioning = "PIT"),
  calibration = list(start = "2020-01-01", end = "2021-12-31",
                     known_at = "2022-01-01", split_id = "synthetic",
                     prediction_origin = "EXTERNAL_TARGET"),
  base_model_version = "synthetic_v1", population_definition = "Hand example",
  outcome_policy = "Fully observed first default", weight_type = "frequency",
  data_origin = "SYNTHETIC")
fit <- calibrate_intercept(c(.1, .1), 2/11, metadata = metadata)
apply_calibrator(fit, c(.1, .2, .5), metadata$target, "synthetic_v1")$value
```

[1.5in]

`calibration_models`
Explicit Calibration, Uncertainty and Model Selection

Description

Native likelihood, pooling, uncertainty and selection methods with frozen artifacts, explicit data roles and retained failed candidates.

Usage

```
calibrate_beta(base_pd, outcomes, weights = NULL, metadata,
  monotonic = TRUE, boundary_policy = "reject", epsilon = 1e-12,
  context = NULL)
calibrate_grade_pd(counts, defaults, grades = NULL, prior_alpha = 0,
  prior_beta = 0, monotonic = TRUE, target_default_rate = NULL,
  default_grade = NULL, confidence = .95, target_weights = NULL,
  context = NULL)
low_default_bound(defaults, counts, confidence = .95,
  dependence = "independent", correlation = 0, pooling = "none",
  integration_tolerance = 1e-8, context = NULL)
```

```
fit_hierarchical_calibration(base_pd, outcomes, groups,
    weights = NULL,
  metadata, variance_bounds = c(1e-8, 25), context = NULL)
calibrate_lgd_level(base_lgd, target_lgd, weights = NULL, metadata,
  transformation = "additive", context = NULL)
estimate_margin_of_conservatism(base_estimate, components,
  confidence = .95, correlation = NULL,
    already_included = character(),
  bounds = NULL, context = NULL)
population_reweighting(sample_features, target_distribution,
  method = "poststratification", cap = NULL,
    outcome_dependent = FALSE,
  context = NULL)
fit_stepwise(X, outcomes, feature_order, candidate_groups,
  mandatory_terms = character(), hierarchy = NULL,
  direction = "bidirectional", criterion = "AIC", estimator = "GLM",
  weights = NULL, max_steps = 100, lr_alpha = .05,
  training = NULL, context = NULL)
fit_regularization_path(X, outcomes, feature_order, penalty_grid,
  split_plan, estimator = "GLM", elastic_net_alpha = 0,
  scoring = "log_loss", weights = NULL, context = NULL)
resample_fit_uncertainty(X, outcomes, feature_order,
    resampling_indices,
  resampling_units, unit_kind, estimator = "GLM", weights = NULL,
  confidence = .95, prediction_X = NULL, selection_spec = NULL,
  context = NULL)
fit_model_selection_protocol(X, outcomes, feature_order, candidates,
  split_plan, metadata, search_budget, scoring = "log_loss",
    context = NULL)
```

Arguments

`base_pd`

Aligned input probabilities; hierarchical fits require interior values.

`outcomes`

Observed binary events for probability methods, or numeric Gaussian responses for OLS/WLS.

`weights`

Positive row weights for selection: unit OLS, inverse-variance WLS, integer frequency GLM. Calibration permits zero frequency weights but requires positive total. LGD also permits real exposure weights when explicitly declared in metadata.

`metadata`

Calibration object with target (event, horizon, time_unit, probability_measure, conditioning), base_model_version, population_definition, outcome_policy, weight_type, data_origin, and calibration (start, end, known_at, split_id, prediction_origin). Empirical predictions must be HOLDOUT or

OUT_OF_FOLD. The full protocol requires HOLDOUT and a window covering actual calibration rows. LGD external targets also permit EXTERNAL_TARGET.

`monotonic`

Constrain beta coefficients a,b nonnegative, or pool ordered grade means monotonically.

`boundary_policy`

Beta input policy: reject exact zero/one, or clip with the declared epsilon.

`epsilon`

Beta clipping threshold in [1e-15,.01].

`context`

Optional calculation context; supplied target fields must match the artifact.

`counts`, `defaults`

Aligned positive integer observation counts and integer event counts between zero and counts. Scalars also accepted by low_default_bound.

`grades`

Unique best-to-worst grade names. Default zero-based string names.

`prior_alpha`, `prior_beta`

Nonnegative beta pseudo-counts per grade; both zero means empirical rates.

`target_default_rate`

Optional performing-grade weighted mean reached by a finite log-odds shift.

`default_grade`

Grade name excluded from performing calibration, required to contain only defaults.

`confidence`

Probability level for intervals or one-sided conservatism; MoC requires at least .5.

`target_weights`

Nonnegative grade weights for target reconciliation only; default counts. Does not change the binomial pooling likelihood.

`dependence`

independent or gaussian_one_factor.

`correlation`

Low-default fixed Gaussian asset correlation in [0,.999], zero for independence; for MoC, a symmetric PSD correlation matrix in component order, default independence.

`pooling`

none or ordered_tail, the latter pooling each ordered grade with all subsequent grades.

`integration_tolerance`

Absolute/relative tolerance for normal-factor integration on [-10,10].

`groups`

Nonempty group IDs, aligned with probabilities and passed again to apply_calibrator.

`variance_bounds`

Positive lower and upper bounds, up to 100, for estimated normal random-intercept variance.

`base_lgd`, `target_lgd`

Loss fractions and desired weighted loss mean. Raw losses may lie outside [0,1] in additive mode.

`transformation`

additive shift, multiplicative scaling of nonnegative losses, or logit shift of strictly interior losses.

`base_estimate`

Scalar risk estimate before explicit conservatism.

`components`

List of objects with unique source_id, category (data, method, general), nonnegative adverse_bias and standard_error.

`already_included`

Source IDs whose uncertainty is already incorporated in the base estimate. Their listed contributions are zeroed.

`bounds`

Optional lower/upper limits for final MoC-adjusted estimate; unbounded value and clipping adjustment remain reported.

`sample_features`

Stratum strings for poststratification, or supplied strictly interior target-membership probabilities for membership_odds.

`target_distribution`

Named target proportions summing to one, or an object containing sample_prior and target_prior for the membership sampling design.

`method`

poststratification or membership_odds.

`cap`

Optional positive cap on raw weights before mean-one normalization; final weights may exceed this cap.

`outcome_dependent`

Explicit declaration of outcome-dependent sampling.

`X`

Complete finite numeric design matrix, with no implicit row omission or global preprocessing.

`feature_order`

Distinct names in exact matrix-column order.

`candidate_groups`

Named list of disjoint feature-name groups; excluded columns cannot enter.

`mandatory_terms`

Candidate group names always retained.

`hierarchy`

Named list mapping group names to required parent groups. Cycles and absent mandatory parents fail.

`direction`

forward, backward, or bidirectional. Backward starts full; other modes start with mandatory groups and intercept.

`criterion`

AIC, BIC, or LR; full likelihood on identical observations.

`estimator`

OLS, WLS, or Bernoulli-logit GLM; estimator and criterion accept either letter case.

`max_steps`

Positive integer limit on accepted search moves.

`lr_alpha`

Nested likelihood-ratio p-value threshold for additions/removals.

`training`

Optional object with ISO known_at and role training_only or development_only. Without it the standalone selection record explicitly reports an unknown cutoff. No temporal guarantee is inferred.

`penalty_grid`

Unique nonnegative lambda values for weighted-average-loss penalization.

`split_plan`

Object with rows and folds. Rows declare entity_id, as_of, label_known_at, role (development, calibration, test). Folds declare train and validation lists of unique zero-based indices. Both languages use zero-based supplied indices.

`elastic_net_alpha`

Zero for ridge, one for lasso, intermediate for elastic net. Intercept is unpenalized.

`scoring`

log_loss or brier for GLM; mse for Gaussian response.

`resampling_indices`

List of complete shared zero-based index arrays, including repetitions, one per replicate.

`resampling_units`

Predeclared cluster or time-block IDs for each original row. Selected units must repeat all constituent rows equally.

`unit_kind`

cluster or time_block; appropriate block length and dependence structure remain user choices.

`prediction_X`

Optional aligned prediction matrix, default original X.

`selection_spec`

Optional stepwise argument object to repeat adaptive selection within each resample.

`candidates`

List of objects with unique id, optional features, penalty_grid and elastic_net_alpha. Full protocol uses GLM candidates.

`search_budget`

Positive integer at least the number of declared candidates; no candidate is silently omitted.

Details

Beta calibration fits plogis(c + a*log(p) - b*log(1-p)); (c,a,b)=(0,1,1) is the identity. Active monotonic constraints are reported: interior inverse-information covariance is not boundary-aware inference.

Hierarchical calibration uses supplied logits as offsets, a global intercept, and independent normal group intercepts. Variance is estimated using Laplace marginal likelihood with analytic profile derivatives; conditional modes are solved individually. Existing groups use their mode. Unknown groups integrate over the fitted population using 40-node Gaussian quadrature. Reported group uncertainty conditions on estimated hyperparameters; it excludes hyperparameter and Laplace-approximation error. Variance-boundary estimates are explicit.

Grade PAVA pools empirical or beta-prior means in grade order. Optional target shifts preserve structural zeros and ones and reject unattainable finite shifts. Intervals are exact binomial without a prior, beta posterior with a prior. They describe pre-target pools conditional on pooling, not post-selection or simultaneous coverage. Zero-default cells retain positive upper uncertainty.

Low-default roots solve P(D <= observed defaults | pd)=1-confidence. For zero independent defaults the answer is 1-(1-confidence)^(1/n). The Gaussian factor mode integrates conditional independent binomial probabilities; ordered-tail pooling also reports monotone upper bounds. Coverage

is conditional on the declared model, not universal under unknown dependence. Integration diagnostics exclude negligible standard-normal probability outside [-10,10].

Additive LGD leaves raw economic losses unclipped. Multiplicative mode requires nonnegative losses and a positive base mean; a zero target ties all predictions. Logit mode requires interior fractions. Exposure weighting is separate from frequency weighting and must be declared explicitly.

MoC is total adverse bias plus qnorm(confidence)*sqrt(s' C s). Component source IDs prevent duplicate counting; already-incorporated components contribute zero. This declared normal uncertainty policy is not an automatic regulatory default. Poststratification rejects missing positive target strata. Membership odds are q/(1-q)*sample_prior/target_prior; externally supplied propensities do not prove covariate overlap. Clipping occurs before normalization; effective sample size and achieved stratum balance expose its effect.

Stepwise Gaussian likelihood includes residual variance and, for WLS, the weight determinant. AIC/BIC use identical rows; LR compares nested likelihoods and records degrees of freedom. Ties within 1e-10 use lexical action/group order; visited states prevent cycling. Rank deficiency, separation and zero residual variance are failures, retained in candidate evidence. No adaptive selected-model fixed- model inference is claimed.

Regularization minimizes weighted average negative log likelihood (Gaussian half squared error) plus lambda*((1-alpha)*sum(beta^2)/2+alpha*sum(abs(beta))). Each fold scales using only its training data; final refit uses development only. Every fold must succeed; exact loss ties prefer stronger penalty. Entity sets are disjoint across fold and stage boundaries, and earlier labels must be known strictly before later observations. Supplied data relationships remain a user responsibility. Standalone selection scoring is intercept plus centered/scaled matrix times coefficients, followed by plogis for GLM. Preserve feature order, training cutoff and the content hash when replaying a SelectionModel.

Resampling repeats complete declared clusters/time blocks and retains every failure. Optional selection repeats within each replicate. Percentile intervals use successful replicates and require at least two; a large failure proportion can invalidate their interpretation. The full protocol performs inner development selection, final development refit, then independent logistic calibration with score tolerance 1e-8. Test labels and predictions never enter optimization. Synthetic engineering evidence is not validation of performance on bank data.

Value

CalcResult. Fitted transformations return a hashed CalibrationArtifact; apply_calibrator validates its target, version, schema, domain and content. Other methods return their estimates and complete diagnostics. Selection methods return a hashed SelectionModel with explicit scoring parameters, training role/cutoff and diagnostic hash. The full protocol additionally returns a common selected_ frozen_model for predict_pd, a separate calibration artifact, all candidate evidence, and untouched test indices; it reports no test metrics.

References

The lme4 authors, glmer documentation.https://lme4.github.io/lme4/reference/glmer.html. Friedman, Hastie and Tibshirani, regularization paths and elastic-net penalties. https://www.jstatsoft.org/article/view/v033i01.

Kull, Silva Filho and Flach (2017), Beta calibration. https://proceedings.mlr.press/v54/kull17a.html. Pluto and Tasche, Estimating Probabilities of Default for Low Default Portfo-

lios. https://arxiv.org/abs/cond-mat/0411699. R stats, step. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/step.html.

Examples

```
low_default_bound(0, 100)$value$bounds
calibrate_grade_pd(c(100, 100, 100), c(1, 4, 2))$value$grades
population_reweighting(c("A", "A", "A", "B"), list(A=.5, B=.5))$value
```

[1.5in]

`classification_bayesian`
Proper-prior Bayesian logit and probit posterior simulation

Description

Native multi-chain Metropolis estimation using explicit shared innovations, frozen posterior draws, uncertainty and actual chain diagnostics.

Usage

```
fit_bayesian_pd(X_train, y_train, feature_order, link, prior_mean,
  prior_sd, sampler_spec, group_spec = NULL, credible_level = .95,
  target, training, model_id, model_version, context = NULL)
```

Arguments

`X_train`, `y_train`

Finite numeric matrix and observed numeric zero/one labels. Single-class training samples are permitted under proper priors.

`feature_order`

Distinct ordered feature names. Features are already prepared; this estimator does not learn scaling.

`link`

logit or probit.

`prior_mean`, `prior_sd`

Finite normal prior means and positive finite SDs for intercept followed by slopes.

`sampler_spec`

List with initial_positions (chain by parameter matrix), lower positive-diagonal proposal_cholesky, normal_innovations (chain by step by parameter array or nested lists), uniforms (chain by step matrix, values strictly in (0,1)), integer warmup, positive thin, and innovation_contract=INDEPENDENT_STANDARD_NORMAL_AND_UNIFORM. At least two chains and eight retained draws required.

`group_spec`

Optional list of distinct observed levels, per-row labels and positive prior_sd for zero-mean group intercepts. The SD is fixed, not inferred. Unknown prediction groups are rejected.

`credible_level`

Interior probability for equal-tail latent-PD credible intervals.

`target`

Full event/horizon/units/probability-measure/conditioning contract.

`training`

Training fingerprint, split_id and ISO known_at information cutoff. Caller supplies mature labels and valid temporal/group splits.

`model_id`, `model_version`, `context`

Immutable model identifiers and optional calculation context.

Details

The posterior combines the Bernoulli likelihood with proper independent normal priors. Symmetric random-walk Metropolis uses caller-supplied independent Gaussian and uniform draws; no proposal adaptation or hidden random seed is used. Retained posterior draws, proposal, initial positions and innovation fingerprint are frozen for audit and prediction. Group intercepts implement partial pooling conditional on their fixed proper prior SD. Parameter order is intercept, slopes, then declared group levels.

Diagnostics include chain acceptance rates, rank-normalized and folded split-Rhat, Geyer initial-positive/monotone paired-sequence ESS for bulk/tails, and raw-quantity mean ESS. ESS is capped by actual draw count. Splitting uses equal first/last halves; an odd middle draw is omitted. Constant chains report unavailable Rhat and zero ESS. DIAGNOSTICS_PASSED requires every coefficient Rhat<=1.01 and bulk/tail ESS>=400. Otherwise a visible failed/insufficient status accompanies the draws; no convergence claim is made. Finite-chain diagnostics cannot prove convergence.

predict_pd averages probabilities over posterior draws, rather than applying the link to mean coefficients. Diagnostics contain equal-tail PD intervals, latent-PD posterior SD, mean-PD Monte Carlo standard error, prediction-specific chain diagnostics and Bernoulli predictive variance. Intervals and MCSE require representative stationary posterior draws, so inspect convergence status before use. These are intervals for conditional PD, not fractional outcome prediction intervals. For grouped fits prediction X is list(X=numeric_matrix,groups=labels); all groups must have been trained.

Prior sensitivity, especially with few defaults, requires refitting explicit substantive alternative priors. No heldout data select a prior implicitly. Model-based posterior uncertainty is conditional on likelihood, priors and independent-row assumptions; it does not establish empirical calibration or independence of repeated borrower forecasts.

Value

FitResult with a data-only ModelArtifact containing all retained posterior draws and chain diagnostics. Use predict_pd with the exact frozen target and features.

References

Vehtari et al. (2021), Rank-normalization, folding, and localization: An improved R-hat for assessing convergence of MCMC. Stan reference manual, Posterior Analysis and Effective Sample Size.

See Also

predict_pd,predict_model_score,serialize_fitted_model

[1.5in]

classification_flexible
Constrained logistic spline models and portable tree ensembles

---

Description

Training-only smooth and tree classifiers with explicit constraints, sampling, inner validation and data-only prediction artifacts.

Usage

```
fit_gam_pd(X_train, y_train, feature_order, linear_features,
    smooth_specs,
  variant, max_iter = 2000L, tolerance = 1e-9, target, training,
  model_id, model_version, context = NULL)
fit_random_forest_pd(X_train, y_train, feature_order, sampling_plan,
  max_depth, min_samples_leaf = 1L, min_weight_leaf = 0,
  criterion = "gini", weights = NULL, target, training, model_id,
  model_version, context = NULL)
fit_gradient_boosting_pd(X_train, y_train, feature_order,
    train_row_ids,
  iterations, max_depth, learning_rate,
    monotonic_constraints = NULL,
  l2_regularization = 1, min_child_weight = 1, min_split_loss = 0,
  max_bin = 256L, inner_validation = NULL,
    early_stopping_rounds = NULL,
  target, training, model_id, model_version, context = NULL)
```

Arguments

X_train, y_train

Finite numeric training matrix and observed numeric zero/one labels. Both training classes required.

feature_order

Distinct ordered feature names, frozen for prediction.

linear_features

Names represented linearly in the additive logistic predictor.

`smooth_specs`

List of terms: feature, two boundary_knots, increasing interior_knots, degree 1–3, nonnegative smoothing, monotonicity none/increasing/decreasing and extrapolation error/constant/linear. Each feature occurs exactly once as linear or smooth.

`variant`

unconstrained or monotone. Monotone requires at least one truly constrained smooth.

`max_iter`, `tolerance`

Positive iteration limit and convergence tolerance below one.

`target`, `training`

Event/horizon/units/measure/conditioning target and training fingerprint, split_id, known_at information cutoff. Caller supplies mature labels and temporal/group split design.

`model_id`, `model_version`, `context`

Immutable model identifiers and optional calculation context.

`sampling_plan`

List of trees with sample_indices (zero-based training row draws, repeats count) and node_features: named list mapping root/L/R/LL/etc. to ordered distinct zero-based candidate feature indices. Every eligible visited node must have a draw.

`max_depth`

Maximum tree depth; forest permits integers 1–20.

`min_samples_leaf`, `min_weight_leaf`

Minimum sampled row count and total positive case weight per forest leaf.

`criterion`

gini or log_loss for weighted CART impurity.

`weights`

Nonnegative forest case weights with positive support in both training classes.

`train_row_ids`

Distinct nonempty training identifiers, disjoint from supplied inner-validation rows.

`iterations`, `learning_rate`

Positive boosting round count and shrinkage in (0,1].

`monotonic_constraints`

Optional vector of -1/0/1, one constraint per input feature.

`l2_regularization`, `min_child_weight`, `min_split_loss`

Nonnegative XGBoost leaf penalty, minimum child Hessian weight and minimum split gain.

`max_bin`

Histogram bin count, integer at least two.

`inner_validation`

Optional list X/y/row_ids/role/split_id/known_at. Role must be inner_validation, split distinct from training, IDs disjoint, cutoff no later than model training cutoff. One-class validation allowed.

`early_stopping_rounds`

Optional positive patience, requiring inner_validation. Only the best evaluated tree prefix is retained.

Details

GAM minimizes mean Bernoulli negative log likelihood plus half the smoothing-weighted squared second differences of full spline coefficients. The first coefficient of each smooth is zero to identify its intercept. Monotone spline coefficients are cumulative nonnegative increments, with reversed sign for decreasing terms; constraints operate during L-BFGS-B fitting. Frozen tangent/constant/error boundary behavior preserves the declared policy. A native LP detects feasible separating recession directions in the penalty null space at relative eigenvalue tolerance 1e-10, respecting monotone bounds; smoothing alone does not cure affine separation. Such fits are rejected. Projected KKT residual and EDF on free coefficients are reported. No smoothing-selection or active-constraint uncertainty interval is inferred. No smooths reduces to ordinary logistic MLE.

Forests evaluate all valid weighted CART midpoint splits using the caller's bootstrap/subsample and feature draws. Gain ties use candidate order then increasing threshold at tolerance 1e-14. Weighted leaf proportions are averaged without clipping. OOB predictions exclude all trees containing that original row and report coverage, counts and Brier; OOB is not a temporal or obligor-group holdout. Randomness is explicit in the supplied design rather than hidden in package-global state.

Boosting uses native XGBoost CPU hist Bernoulli loss without stochastic subsampling. Initial margin is the logit of training default fraction. Frozen leaf values already include shrinkage; input casting to float32 and strict-less-than routing match backend prediction. Nonrepresentable values are rejected. All evaluated loss rounds and the selected early-stop prefix are recorded. Feature/label availability and grouping independence remain caller responsibilities; disjoint record IDs alone cannot prove temporal integrity. No empirical calibration or uncertainty interval is claimed by these fits.

Value

FitResult with a frozen data-only ModelArtifact and model-specific numerical, design and validation diagnostics. Use predict_pd or predict_model_score with matching target and features.

References

Eilers and Marx (1996), Flexible smoothing with B-splines and penalties. Breiman (2001), Random Forests. Chen and Guestrin (2016), XGBoost: A Scalable Tree Boosting System. R splines::splineDesign documentation and XGBoost model/monotonic-constraint documentation.

See Also

predict_pd, predict_model_score, serialize_fitted_model

[1.5in]

classification_models
Portable probabilistic classification models and explicit raw scores

---

Description

Native training-only estimation with frozen data-only prediction artifacts, explicit target and feature ordering, and model-specific diagnostics.

Usage

```
fit_discriminant_pd(X_train, y_train, feature_order, variant,
  covariance_estimator = "mle", covariance_ridge = 0, priors = NULL,
  target, training, model_id, model_version, context = NULL)
fit_naive_bayes_pd(X_train, y_train, feature_order, feature_types,
    variant,
  alpha = 1, variance_floor = 0, missing_policy = "error",
  unknown_policy = "error", priors = NULL, target, training,
  model_id, model_version, context = NULL)
fit_svm_score(X_train, y_train, feature_order, kernel, C,
    gamma = NULL,
  scale = TRUE, class_weights = NULL, tolerance = 1e-7,
    max_iter = NULL,
  target, training, model_id, model_version, context = NULL)
predict_model_score(fitted_model, X, feature_order, expected_target,
  context = NULL)
fit_firth_logit(X_train, y_train, feature_order,
    fit_intercept = TRUE,
  max_iter = 1000L, tolerance = 1e-9, target, training, model_id,
  model_version, context = NULL)
fit_penalized_glm(X_train, y_train, feature_order, penalty,
    lambda_path,
  selected_lambda, l1_ratio = NULL, fit_intercept = TRUE,
    scale = TRUE,
  weights = NULL, max_iter = 10000L, tolerance = 1e-9, target,
    training,
  model_id, model_version, context = NULL)
fit_ordinal_model(X_train, y_train, feature_order,
    ordered_categories,
  link, default_categories = NULL, max_iter = 1000L,
    tolerance = 1e-9,
  target, training, model_id, model_version, context = NULL)
```

Arguments

`X_train`, `X`

Numeric matrix; naive Bayes also accepts a data frame or list of rows preserving mixed numeric/string types.

`y_train`

Observed numeric zero/one outcomes, or category strings for ordinal fitting. No missing outcomes are treated as nondefaults.

`feature_order`

Distinct ordered feature names, exactly matching the frozen model at prediction.

`variant`

lda/qda for discriminants; gaussian/categorical/mixed for naive Bayes.

`covariance_estimator`

mle or unbiased. LDA pooled scatter divides by n or n-2; QDA class scatter by n_k or n_k-1.

`covariance_ridge`

Nonnegative variance added to each diagonal on the supplied feature scale. Singular covariance without sufficient ridge is rejected.

`priors`

Optional positive class probabilities in nondefault/default order, summing to one; otherwise observed proportions.

`target`, `expected_target`

Full event, horizon, units, probability measure and conditioning contract; expected_target must match exactly.

`training`

Training data fingerprint, split_id, ISO known_at cutoff and optional data_origin. Caller supplies outcomes available by this cutoff.

`model_id`, `model_version`

Immutable model identifiers.

`context`

Optional calculation context.

`feature_types`

gaussian/categorical per feature. Mixed requires both types.

`alpha`

Strictly positive additive categorical count smoothing.

`variance_floor`

Nonnegative addition to Gaussian maximum-likelihood variance.

`missing_policy`

error, marginalize or category. Marginalize omits the feature likelihood factor; category reserves a categorical missing bin and rejects missing Gaussian values.

`unknown_policy`

error or reserved. Reserved adds a distinct unknown-category bin to smoothing denominators.

`kernel`

linear or rbf.

`C`, `gamma`

Positive SVM cost and explicit positive RBF gamma. Linear kernel requires NULL gamma.

`scale`

Fit weighted population standard deviation and center only on training rows; freeze them for prediction. Constant scales become one.

`class_weights`

Optional positive SVM loss weights for nondefault/default classes.

`tolerance`, `max_iter`

Convergence controls. Native e1071 exposes no iteration cap: SVM requires max_iter=NULL and verifies dual KKT residual; non-NULL caps are rejected. Other models require positive integer iteration bounds.

`fitted_model`

Frozen ModelArtifact or FitResult.

`fit_intercept`

Include an intercept, unpenalized for elastic-net models.

`penalty`

ridge, lasso or elasticnet for Bernoulli logistic loss.

`lambda_path`, `selected_lambda`

Unique decreasing nonnegative penalties and exact caller-selected member. No validation selection is performed implicitly.

`l1_ratio`

Interior elastic-net mixing fraction; NULL for ridge/lasso endpoints.

`weights`

Nonnegative case weights, positive total in both classes. Likelihood is normalized by their sum.

`ordered_categories`

Distinct outcome categories ordered from lower to higher credit risk, all represented in training.

`link`

Ordinal logit or probit.

`default_categories`

Optional ordered terminal tail of categories defining default at target horizon. Required before ordinal PD prediction.

Details

Gaussian discriminants apply Bayes' theorem to fitted class means/covariances. Naive Bayes additionally assumes conditional independence, and its Gaussian variances use maximum-likelihood denominators. These posteriors do not establish empirical calibration.

Elastic-net minimizes weighted mean negative logistic likelihood plus lambda times ((1-alpha)/2 times squared slope norm plus alpha times absolute slope norm). Each path point records objective and KKT residual. Penalties operate on fitted standardized slopes when scale is TRUE. Zero penalty checks rank and separation and reproduces ordinary logistic likelihood. No post-selection uncertainty is supplied.

Firth maximizes Bernoulli log likelihood plus one half log determinant of expected Fisher information. Native brglm2 MPL_Jeffreys with a=1/2 is checked against the exact adjusted score. Full rank is required; single-class samples and separation are permitted. Stored inverse expected information is a local diagnostic, not a penalized profile confidence interval.

Ordinal models obey P(Y<=jlx)=F(threshold_j-x beta). Prediction scores expose all category probabilities in diagnostics. Default categories must be an ordered terminal tail. The parallel-slope diagnostic jointly stacks threshold-wise binary scores and retains cross-threshold covariance in an IID sandwich Wald test. Singular or separated diagnostic fits yield NOT_ESTIMABLE; the diagnostic does not prove proportionality. Covariance parameters are slopes, first threshold and log subsequent threshold spacings.

SVM outputs are signed margins, never PDs without independent calibration. Frozen support vectors/dual coefficients/intercept are portable; higher margins mean default. Preprocessing never uses prediction rows. Model-based likelihood uncertainty assumes independent observations; temporal/group holdout selection and mature labels remain explicit caller responsibilities.

Value

Fits return FitResult with a portable ModelArtifact and diagnostics. predict_model_score returns CalcResult containing a numeric score array, explicit scale, orientation and model lineage; ordinal diagnostics include category probabilities.

References

Firth (1993), Bias reduction of maximum likelihood estimates. Kosmidis and Firth (2021), Jeffreys-prior penalty, finiteness and shrinkage in binomial-response generalized linear models. Tay, Narasimhan and Hastie (2023), Elastic Net Regularization Paths for All Generalized Linear Models. Chang and Lin (2011), LIBSVM: A library for support vector machines. MASS package documentation for lda, qda and polr.

See Also

predict_pd, serialize_fitted_model, load_fitted_model

[1.5in]

classification_neural
Native portable MLP, GRU and LSTM credit likelihood models

Description

CPU dense and recurrent neural estimation with actual analytic gradients, explicit Adam orders, frozen weights and causal sequence availability contracts.

Usage

```
fit_neural_pd(X_train, y_train, feature_order, architecture,
  initial_weights, optimizer_spec, train_row_ids, scale = TRUE,
  inner_validation = NULL, target, training, model_id,
    model_version,
  context = NULL)
fit_sequence_pd(sequences, y_train, feature_order, variant,
    hidden_size,
  initial_weights, optimizer_spec, train_row_ids, label_known_times,
  training_as_of_time, include_time_gaps = TRUE, time_gap_scale = 1,
  scale = TRUE, inner_validation = NULL, target, training, model_id,
  model_version, context = NULL)
```

Arguments

X_train, y_train

Finite numeric matrix and observed numeric zero/one labels. Both training classes required.

feature_order

Distinct ordered raw feature names, matching the frozen prediction contract.

architecture

MLP hidden_units and matching activations tanh/relu/sigmoid; empty hidden layers gives a logistic output directly.

initial_weights

Explicit finite numeric parameters. MLP keys layer0_weight/layer0_bias etc., matrices input-width by output-width and one-dimensional biases. Recurrent keys input_weight, hidden_weight, input_bias, hidden_bias, output_weight, output_bias; see Details.

optimizer_spec

algorithm=adam, positive learning_rate and batch_size, explicit epoch_orders (complete zero-based training permutations). Optional beta1=.9, beta2=.999, epsilon=1e-8, l2=0, positive gradient_clip_norm and early_stopping_patience. Patience requires inner_validation.

`train_row_ids`

Distinct nonempty record identifiers disjoint from inner validation. Caller additionally controls borrower grouping.

`scale`

Train-only population mean/SD scaling. Sequence scaling pools only retained available training observations. Constant SD becomes one.

`inner_validation`

Explicit role=inner_validation, distinct split_id, disjoint row_ids, ISO known_at no later than training cutoff; X/y for MLP or sequences/y/label_known_times for sequences. Does not silently join training.

`target`

Event, positive horizon, units, probability measure and conditioning. Sequence horizon uses the same declared numeric clock unit.

`training`

Fingerprint, split_id and ISO known_at information cutoff.

`model_id`, `model_version`, `context`

Immutable model identifiers and optional calculation context.

`sequences`

List of values matrices, observation_times, known_times, numeric zero/one mask and forecast_time. At least one retained available observation per sequence, strictly increasing in time. All numeric padding must be finite.

`variant`, `hidden_size`

gru or lstm, and arbitrary positive hidden width for one unidirectional recurrent layer.

`label_known_times`, `training_as_of_time`

Numeric label knowledge times and model data cutoff on the sequence clock. Labels cannot predate forecasts or exceed cutoff; nondefaults require full-horizon maturity.

`include_time_gaps`, `time_gap_scale`

Append first gap zero and subsequent retained observation gaps divided by a positive fixed scale.

Details

Forward propagation and full backpropagation are native, with no Python bridge or runtime neural backend. Loss is mean Bernoulli negative log likelihood. Adam bias-corrected moments use a coupled L2 weight-gradient penalty, not AdamW; biases are unpenalized. Optional global gradient clipping occurs after L2. Initialization fingerprints, epoch orders, update counts and full loss paths are recorded. Completion of an epoch budget is not called global optimization convergence.

For input width p and hidden width h, GRU input/hidden weights have shapes (p,3h)/(h,3h), LSTM (p,4h)/(h,4h). Gate-width input and hidden biases are separate. Output weight is (h,1), output bias

length one. GRU gate order r,z,n uses reset after the hidden affine: r=sigmoid(input_r+hidden_r), z=sigmoid(input_z+hidden_z), n=tanh(input_n+r*hidden_n), h=(1-z)*n+z*previous_h. LSTM i,f,g,o uses c=f*previous_c+i*g, h=o*tanh(c), sigmoid i/f/o and tanh g. These matrices are transposes of the corresponding PyTorch stored gate matrices. Recurrent and cell states reset to zero for every record and prediction call. Genuine gated BPTT updates all recurrent parameters.

Only mask-one observations whose observation and knowledge times do not exceed forecast time enter sequence state or scaling. Known time cannot precede observation. Padding/future/late-known observations are excluded before processing. At least one observation must remain; retained times are strictly ordered. Time-gap augmentation increases input width by one. Pooling weights each retained sequence observation equally, so longer sequences contribute more to scaling. Label-known times prevent immature negatives, while observed early positive defaults may enter earlier. Correct event evidence and calendar/unit interpretation remain caller responsibilities.

With inner validation, the frozen artifact retains the best validation epoch, including epoch zero when all updates harm validation. Without validation it retains the final epoch. Early stopping never reads an undeclared final test set. Disjoint row IDs do not prove independence of repeated borrower records. No posterior uncertainty or empirical probability calibration is inferred from neural training loss.

Value

FitResult containing only portable architecture, weights, preprocessing, target and training/optimization diagnostics. Use predict_pd or predict_model_score with matching features and target; sequence prediction receives the same sequence-record structure.

References

Kingma and Ba (2015), Adam: A Method for Stochastic Optimization. PyTorch GRU and LSTM documentation for gate equations and parameter conventions.

See Also

`predict_pd`,`predict_model_score`,`serialize_fitted_model`

[1.5in]

`collections`
Dated Collections Episodes, Operational Cure and Recovery Economics

Description

Construct collections histories from explicit dated states and coverage, preserve censored cure and redefault outcomes, separate pre-action information from treatment stages, and evaluate local repayment/recovery scenarios.

Usage

```
build_collections_episodes(states, actions, coverage, episode_rule,
  evaluation_cutoff, context = NULL)
build_cure_targets(episodes, horizon_days, redefault_horizon_days,
  evaluation_cutoff, context = NULL)
build_collections_action_history(actions, features, outcomes,
    coverage,
  index_stage, required_features, horizon_days,
    maximum_feature_age_days,
  evaluation_cutoff, context = NULL)
simulate_repayment_arrangement(contract, baseline, candidate,
  affordability_paths, discount_factors, period_end_dates, currency,
  context = NULL)
expected_collections_value(actions, discount_factors, ead, currency,
  period_end_dates, context = NULL)
```

Arguments

`states`

Data frame or list of rows with unique state_event_id, account_id, state, occurred_at and known_at. These are complete state changes, with a baseline at or before observed_from, not intermittent state samples. One state per account and occurrence date; simultaneous states require upstream resolution.

`actions`

For episode/history construction, rows with unique action_id, account_id, action_type, stage, occurred_at and known_at. Stage is ASSIGNED, OFFERED, ACCEPTED, PERFORMED or CANCELLED. Action history additionally requires a treatment_id joining stage events for one account/action type, with at most one record per stage. For expected_collections_value, action scenario rows are described in the economic inputs section below.

`coverage`

Rows with unique account_id, observed_from, observed_through and known_at. Coverage is an independent assertion of complete capture between the inclusive dates; known_at cannot precede observed_through. Unknown-at-cutoff coverage is not used to establish observation completeness.

`episode_rule`

Named list with rule_id, version, problem_states, clean_states, terminal_states, default_states, probation_days and separation_days. State sets are unique strings; problem and clean sets must be nonempty and mutually disjoint from terminal states. Default states are a subset of problem states. Durations are explicit nonnegative integer calendar days.

`evaluation_cutoff`

Inclusive ISO calendar date. Events must both occur and be known by this date; earlier occurrence alone does not establish availability.

`context`

Optional named calculation context or context result. Economic methods reject a conflicting context currency.

`episodes`

Result/value from build_collections_episodes, with canonical history_hash and episode_hash values intact. Rebuild at a new cutoff.

`horizon_days`

Positive integer calendar days from episode start or the actual indexed action occurrence.

`redefault_horizon_days`

Positive integer calendar days after sustained cure.

`features`

Rows with unique feature_record_id, account_id, feature, value, observed_at and known_at. A value key is required, and NULL is permitted to express missing information. Tied latest observation/knowledge dates are rejected.

`outcomes`

Rows with unique outcome_id, account_id, event_type, occurred_at and known_at. Outcomes on or before index date remain an audit list, not prospective outcomes. Later known events up to the horizon end are retained.

`index_stage`

ASSIGNED or OFFERED. No action index is inferred from eventual acceptance, performance or successful recovery.

`required_features`

Nonempty vector/list of distinct required feature names.

`maximum_feature_age_days`

Nonnegative integer age limit at the index date. Older latest available features are retained but flagged stale.

`contract`

Named list with currency, principal, principal_arrears, accrued_interest, fees and as_of. Amounts are finite nonnegative scalars. Arrears principal is a subset of principal and is never added again to the claim.

`baseline`, `candidate`

Explicit plan specifications, each with plan_id, payments, period_interest_rates, new_fees, servicing_costs, receipt_fractions and logical capitalize_unpaid_interest. Numeric vectors have one entry per period; all are nonnegative and receipt fractions are at most one.

`affordability_paths`

Rows with unique path_id, currency and finite numeric available_cash vector per period. Negative available cash is retained.

`discount_factors`

Nonnegative finite numeric vector, one factor per period. No interest-rate or discount convention is inferred from the factors.

`period_end_dates`

Nonempty vector/list of strictly increasing ISO dates. Repayment period ends must follow the contract as_of date.

`currency`

Explicit nonempty currency, matching contract/budget/action rows and any context currency. No conversion is inferred.

`ead`

Finite nonnegative scalar exposure denominator. At zero EAD the expected-net-recovery ratio is NULL; monetary recovery remains available.

Value

Each method returns a CalcResult with explicit limitations.

build_collections_episodes returns episodes, complete cutoff-filtered state_history, unlinked_actions, rule, coverage, evaluation_cutoff and history_hash. Episodes contain identifiers, start/end, OPEN/CURED/TERMINATED status, previous_episode_id, timeline, actions, ever_default, observation_cutoff, cure_candidate_at and episode_hash. Histories and rules are bound by canonical hashes.

build_cure_targets returns one row per episode with horizon_end, cure_label/status, cure_at, redefault_label/status, redefault_at, ever_default and cutoff. Labels are 1 or 0 only when their corresponding event or non-event is established; otherwise NULL with censoring status.

build_collections_action_history returns rows and unindexed_treatments. Rows contain preaction_features, feature_sources, missing/stale features, stage_dates, invalid_stage_order, descriptive_eligible, accepted/performed indicators, prior_or_same_day_outcome_ids, subsequent outcomes, outcome_complete and row_hash. causal_effect_identified is always FALSE.

simulate_repayment_arrangement returns baseline/candidate ledgers, NPV change, principal_arrears_included, contract_changed=FALSE and default_status_changed=FALSE. Each plan contains schedule, discounted net-cashflow NPV, remaining_claim and household residual cash/shortfall dates.

expected_collections_value returns one row per supplied action with expected_net_recovery, its EAD ratio, positive-mass scenario min/max, expanded accepted/declined scenarios, support_status, assumption_reference and causal_effect_identified=FALSE.

Episode and cure boundaries

An episode starts in a declared problem state. A clean transition starts a quiet candidate interval. Episode separation requires complete coverage through max(probation_days, separation_days). A problem event exactly on that boundary interrupts the interval. Terminal states close the episode as TERMINATED. An open data endpoint, a single payment or an unobserved gap never establishes cure. A later problem spell carries the previous episode identifier. Historical default is not erased by operational cure.

The cure target uses probation_days, independently of the longer possible episode separation duration. A supported cure on or before the episode horizon end is a positive observed event, even if later follow-up is censored. A negative label requires coverage from episode start through horizon end and a mature cutoff. A terminal exit within the target window censors a non-cured episode.

Redefault uses a separate risk set starting strictly after cure through the inclusive post-cure horizon end. The first default versus terminal event determines positive redefault versus exit censoring. A negative label requires complete post-cure coverage. The full sealed state history retains terminal events after an episode has already closed, so they cannot disappear from follow-up.

Action information and interpretation

Features must both be observed and known strictly before the actual assignment or offer date. Same-day feature records are excluded because intraday ordering is not established by calendar dates. Eligible records are selected by account/name, ordered by observation date, knowledge date and record identifier. Late logging of the action does not move its information cutoff. Missing and stale values remain visible; no imputation is performed. Duplicate latest observation/knowledge timestamps require explicit resolution.

ASSIGNED, OFFERED, ACCEPTED and PERFORMED dates are checked for chronological order; invalid order makes a row descriptively ineligible. CANCELLED remains a separate stage record. A later accepted/performed status is a post-index outcome of the process, not a pre-index feature. Treatments without the declared index stage are listed separately. Same-day outcomes have unresolved intraday order and are never used as prospective outcomes. Outcome absence is not a zero label when coverage is incomplete. No causal effects, treatment eligibility or action ranking are inferred from these factual groups.

Economic inputs and cash reconciliation

For repayment simulation, interest is an effective rate for each supplied period on its opening principal. It is not an annual rate and receives no hidden day-count conversion. New fees and interest enter one claim ledger. Scheduled cash is capped by the outstanding claim; receipt fractions determine actual cash. The waterfall pays fees, then interest, then principal. Remaining interest is capitalized only when explicitly selected. Unused payment capacity, all waterfall amounts and the closing claim remain visible.

NPV sums received cash less servicing cost after applying period factors. Remaining claims carry no assumed terminal recovery. Household residual cash subtracts scheduled obligations, not the lower cash receipts under assumed nonpayment. Plans do not change a legal contract or a recorded default status.

For expected_collections_value each action row requires action_id, currency, uptake_probability, accepted_scenarios, declined_scenarios, offer_cost, support_status and assumption_reference. Each branch must contain a complete conditional scenario distribution, including when branch

probability is zero. A scenario contains unique scenario_id, probability and cashflows. Conditional probabilities lie in [0,1] and sum to one within 1e-12; they are used as supplied. Each cashflow has unique cashflow_id within its scenario, kind RECOVERY or COST, period_end matching a declared date, and finite nonnegative amount.

Joint scenario masses are uptake times accepted conditional mass, or one minus uptake times declined conditional mass. Recovery and servicing costs are discounted per scenario. Offer cost occurs once at time zero; it must not also appear as a scenario cost. Expected recovery weights these exclusive net values. Cashflow IDs prevent repeated records within one scenario; callers must give the same economic cashflow one identifier. Scenario ranges are not confidence intervals and inputs are conditional assumptions, not identified treatment effects.

Examples

```
plan <- list(plan_id = "base", payments = c(50, 50),
  period_interest_rates = c(0, 0), new_fees = c(0, 0),
  servicing_costs = c(0, 0), receipt_fractions = c(1, 1),
  capitalize_unpaid_interest = FALSE)
contract <- list(currency = "EUR", principal = 100, principal_arrears = 20,
  accrued_interest = 0, fees = 0, as_of = "2024-01-01")
simulate_repayment_arrangement(contract, plan, plan, list(), c(1, 0.95),
  c("2024-02-01", "2024-03-01"), "EUR")$value$npv_change
```

[1.5in]

`collections_backtest`
Validate Dated Collections Cashflow Forecasts

Description

Factual fixed-horizon assessment of frozen repayments, recoveries, costs and recovery timing with explicit historical knowledge and complete cash-capture coverage.

Usage

```
collections_backtest(forecasts, episodes, cashflows, actions,
  cashflow_coverage, discount_contract, design, uncertainty,
  cluster_resamples = NULL, confidence_level = .95, context = NULL)
```

Arguments

`forecasts`

Common validation forecast records plus account, sealed episode, index action and dated predicted cashflows.

`episodes`

Sealed history from `build_collections_episodes` at the evaluation cutoff.

`cashflows`

Unique dated episode cashflows with explicit REPAYMENT, RECOVERY or COST kind, nonnegative amount, currency and knowledge date.

`actions`

Historically known action records; selected assignment or offer must match the sealed episode record.

`cashflow_coverage`

Per-episode complete cash-capture intervals with knowledge dates.

`discount_contract`

Explicit historically known flat effective annual rate greater than minus one, ACT_365F day count, matching currency and reference.

`design`

Frozen model, parameter, split, population, currency, target and evaluation-cutoff identities.

`uncertainty`

`none` or `cluster_bootstrap`.

`cluster_resamples`

Supplied shared original-cluster bootstrap draws.

`confidence_level`

Percentile coverage strictly between zero and one.

`context`

Optional calculation context consistent with the design.

Details

Amounts are discounted to forecast date by the effective annual rate over actual days divided by 365. Bias is prediction minus observation. Forecast and actual dates use an inclusive forecast-to-horizon window. Only mature horizons with complete cash capture and positive evaluation weight are scored. Late-known cash is excluded; state-history coverage alone cannot establish complete cash capture. Open episodes stay open. Recovery timing is recovery-amount-weighted days, defined only for positive forecast and observed recovery mass. Bootstrap intervals condition on frozen forecasts and become non-estimable when any supplied resample lacks the required outcome. Action-group metrics describe factual assignment cohorts, not causal effects.

Value

A CalcResult containing a ValidationResult, amount and timing metrics, coverage, uncertainty and episode/action/cashflow traces. It does not grant release or empirical approval.

[1.5in]

`compile_credit_decision_policy`
Compile, evaluate and explain local credit decision policies

Description

Deterministic rules keep probability estimates, affordability, eligibility and operational information separate. Evaluation returns a local recommendation with fingerprints and a complete rule trace.

Usage

```
compile_credit_decision_policy(specification, context = NULL)
apply_credit_decision_policy(policy, risk, affordability,
    eligibility,
  operations, as_of, known_at, context = NULL)
decompose_policy_decision(decision, reason_dictionary,
    context = NULL)
```

Arguments

`specification`

Named policy list with schema version, identifier, version, approval reference, knowledge and effective dates, physical risk target, typed source fields, explicit default and missing actions, and rule array.

`policy`

Compiled policy result or its fingerprinted value.

`risk`

Named list with risk target, freshness, observation and knowledge dates and risk fields required by the policy.

`affordability`

Separate named list of dated affordability evidence and status.

`eligibility`

Separate named list of dated eligibility evidence and status.

`operations`

Separate named list of dated operational evidence and status.

`as_of`

Decision date in ISO YYYY-MM-DD format.

`known_at`

Knowledge cutoff, not before the decision date.

`decision`

Previously evaluated decision result or its fingerprinted value.

`reason_dictionary`

Named list mapping every matched rule's reason code to a supplied explanation. Unmatched rules need no description.

`context`

Optional calculation context.

Details

Rules have a unique rule_id, nonnegative integer priority, action, reason_code and a list of conjunctive predicates. Lower priorities execute first; equal priorities sort by rule identifier. Predicates name a declared source field and use eq, ne, lt, le, gt or ge. Categories allow equality operators only. Numeric thresholds are finite binary64 values. Contradictory rules and overlapping equal-priority rules with different actions are rejected. Lower-priority overlap and shadowing are reported as diagnostics.

The policy effective interval includes its start and excludes its end. Missing, future, stale or incompatible information takes the explicit missing action before the rule result. Under specification 1.0, approval also requires AFFORDABLE, ELIGIBLE and CLEAR statuses from the three separate nonrisk sources; otherwise it becomes REFER. Gate evaluation never alters a probability. Default action cannot be APPROVE. Approval references record caller assertions and do not establish institutional authorization. No credit, payment, limit or price is executed.

Specification 1.1 requires approval_contract with exactly domain, affordability and required_gates. Domains are RETAIL_CREDIT, WHOLESALE_CREDIT, INVESTMENT and COUNTERPARTY_LIMIT. Affordability mode REQUIRED preserves AFFORDABLE. Only investment/counterparty domains permit NOT_APPLICABLE, with nonempty reason and reference and nonnegative whole-day maximum_age_days. The dated runtime affordability object must then declare NOT_APPLICABLE, the same domain, matching applicability_reference and an exclusive valid_until date. The age limit is inclusive. Missing, stale, future-known, expired or differently bound evidence prevents approval.

Additional gates are distinct named lists with field and value, referencing an explicitly declared categorical field and one allowed category. All gates must pass together. Mandatory ELIGIBLE, CLEAR, current risk and target checks remain independent of rule priority. Failed approval status gates yield REFER. The decision adds approval_domain and approval_gate_trace. Empty gate arrays are permitted but do not waive mandatory gates. Contract domains and legal applicability conclusions are supplied assertions, not inferred approvals.

Specification 1.0 rejects approval_contract, retains its existing hashes and behavior, and cannot silently ignore the new semantics. Artifact envelopes remain schema 1.0; the nested specification carries version 1.1.

Array fields, including category values, rules and predicate lists, must be unnamed lists even for one or zero elements. This preserves canonical JSON array identity across native R and Python. Objects must be named lists. Fingerprints detect content changes; they are not digital signatures.

Value

A CalcResult. Compilation returns a DecisionPolicy artifact; application returns a DecisionArtifact containing action, findings, rule trace, input and policy hashes. Decomposition returns recorded matched reasons and identifies which rule was selected, without recalculating a decision.

[1.5in]

`compute_credit_ale`
Supported First and Purified Second-Order Accumulated Local Effects

---

Description

Weighted finite-bin local differences with explicit interpolation, empirical centering and unsupported-cell reporting.

Usage

```
compute_credit_ale(predictor, X, model_contract, selected_features,
  bins, feature_types, weights = NULL, min_bin_count = 1,
    context = NULL)
```

Arguments

`predictor`

Frozen local prediction function taking a numeric row matrix.

`X`

Reference feature matrix or row lists in frozen feature order.

`model_contract`

Frozen model identity, feature order, output scale and target.

`selected_features`

One or two distinct selected feature names.

`bins`

Named list of finite strictly increasing edges for each selected feature, covering every reference value.

`feature_types`

Named type declarations; selected features must be continuous.

`weights`

Optional nonnegative reference weights with positive total.

`min_bin_count`

Positive minimum count of positive-weight rows in every required bin/cell.

`context`

Optional common calculation context.

Details

Bins are left-closed/right-open except the final upper endpoint. First-order weighted upper-minus-lower local differences are accumulated, linearly interpolated within bins and centered over the original weighted reference. Second-order four-corner mixed differences are double-accumulated; both first-order accumulated components are removed, followed by weighted empirical centering of the bilinear surface. The purification components are returned. Empty or insufficient cells yield NON_ESTIMABLE_SUPPORT with NULL global curves, while supported local differences remain available. No gaps are filled with zero effects or interpolated across. Numeric categories require a separate methodology and are rejected unless explicitly declared continuous. Counts and correlation do not establish off-manifold boundary support or causal effects. The exact finite-bin interpolation convention is part of the method contract.

Value

CalcResult with centered boundary grid and reference effects when supported, local differences, counts, masses, unsupported cell indices, purification components, correlations and hashes.

References

Apley and Zhu, Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. https://arxiv.org/abs/1612.08468.

[1.5in]

`compute_credit_ice`
Feasible Ordered Local Credit Prediction Paths

Description

Evaluate an explicit finite ICE path while retaining infeasibility, nonmonotonicity, scalar crossings, optional rating classes and full-policy replays.

Usage

```
compute_credit_ice(predictor, record, grid, model_contract,
  feasibility_contract, varying_features, thresholds = NULL,
  rating_scale = NULL, policy_evaluator = NULL,
    policy_contract = NULL,
  context = NULL)
```

Arguments

`predictor`

Frozen local function returning one prediction per numeric input row.

`record`

Numeric baseline feature vector.

`grid`

Ordered numeric matrix or list of rows in varying_features order.

`model_contract`

Frozen attribution model contract with ordered features, output scale and explicit target.

`feasibility_contract`

contract_id, version, finite bounds for all features, actionable_features and immutable_features partition, integer_features, one_hot_groups, linear_constraints and derived_relations. Linear records have constraint_id, coefficients, operator and rhs. Derived records have constraint_id, output_feature, two input_features, and product or ratio operation. Empty lists are explicit.

`varying_features`

Ordered names of grid columns.

`thresholds`

Optional records with threshold_id, value and operator <= or >=.

`rating_scale`

Optional versioned rating scale for probability output only.

`policy_evaluator`

Optional local function(record, prediction) returning decision (sealed DecisionArtifact) and explained_input_hash of the list containing record and prediction. Record is a plain ordered list.

`policy_contract`

policy_hash, source_mode REPLAY_FULL_POLICY and fixed_source_hashes named list identifying any unchanged risk, affordability, eligibility or operations inputs.

`context`

Optional common calculation context.

Details

Bounds, immutable features, integers and one-hot identities are checked before prediction. Linear and derived identities use relative tolerance 1e-10; zero-denominator ratios fail. Infeasible points remain NULL with findings and no predictor or policy invocation. Gaps are never interpolated. Turns are along the supplied ordered path and do not imply a sorted feature axis or a unique continuous threshold. Rating classes use map_pd_to_grade; policy decisions require a separately bound

full-policy replay. Content binding is a caller assertion, not authentication. Feasibility is declared structural feasibility, not proof of empirical support or real intervention success. No actual credit action is executed.

Value

CalcResult with grid records, predictions, findings, threshold states, adjacent changes, turns, optional grades and sealed decisions, and contract hashes.

See Also

`map_pd_to_grade`, `apply_credit_decision_policy`

[1.5in]

`content_fingerprint`
Canonical Content Fingerprint for Local Data Contracts

Description

Computes the SHA256 canonical identifier used by local frozen simulation recipes, parameter snapshots and artifact bindings.

Usage

```
content_fingerprint(value, context = NULL)
```

Arguments

`value`

Finite JSON-compatible data. Use named lists for objects, unnamed lists for arrays and NULL for null. A length-one R atomic vector is a scalar; wrap it in an unnamed list to express a one-element array.

`context`

Optional calculation context.

Details

Object keys must be nonempty printable ASCII and are sorted; string values are UTF-8. Numbers use finite binary64 encoding, with negative zero normalized to zero; equivalent integer and floating values share a digest. Booleans, numbers, strings, arrays, objects and null remain distinct. R named empty lists represent empty objects, while list() is an empty array. The identifier is not a digital signature, authorship proof, or financial schema validation. Exclude an artifact's own digest field before hashing its payload.

Value

A CalcResult whose value is a hexadecimal SHA256 string.

Examples

```
content_fingerprint(list(principal = 100, currency = "EUR"))$value
```

[1.5in]

`counterparty_exposure_backtest`
Validate Frozen Counterparty Exposure and Closeout Forecasts

Description

Physical per-netting-set exposure quantile and amount validation with separate mature observed-default closeout assessment.

Usage

```
counterparty_exposure_backtest(forecasts, observations,
    quantile_levels,
  design, uncertainty, cluster_resamples = NULL,
  confidence_level = .95, context = NULL)
```

Arguments

`forecasts`

Dated common validation records plus frozen counterparty/netting-set identities, legal/CSA versions, sealed netting/collateral paths, optional scenario weights and joint default-event closeout contracts.

`observations`

Matching observed or censored endpoint records with actual path artifacts, knowledge dates, reported exposure and optional observed default closeout.

`quantile_levels`

Distinct probabilities strictly between zero and one.

`design`

Frozen model, parameter, data, population, split, currency, physical horizon and evaluation-cutoff contract.

`uncertainty`

`none` or `cluster_bootstrap`.

`cluster_resamples`

Caller-supplied shared whole-cluster bootstrap draws.

`confidence_level`

Interior percentile confidence level.

`context`

Optional calculation context consistent with the design.

Details

Quantiles use the weighted empirical inverse CDF; exceptions are strictly greater than the threshold. Atom-dependent model tail mass is reported separately from the nominal tail probability. Market observations require exact horizon alignment and known information. Frozen netting/CSA content is compared, and reported exposure reconciles to the exact per-set value without cross-set offset.

Default-conditioned closeout is separately calculated using the public frozen-collateral-unit routine and declared joint market/default path events. Only mature known closeouts are scored; missing defaults are not zero-loss observations. A mature closeout can remain applicable when a horizon market value is censored. Bootstrap intervals condition on the frozen paths and become non-estimable if a resample lacks required default outcomes.

Value

A CalcResult with a ValidationResult, market exposure and quantile scores, separate conditional closeout errors, coverage, uncertainty and reconciliation findings. Reconciliation conflicts fail the declared contract; performance approval is not inferred.

References

Basel Committee on Banking Supervision, CRE50, publication 2024-07-05, definitions 50.15, 50.19 and 50.22–50.28. https://www.bis.org/committees/bcbs/basel-framework/standard/cre/50/inforce/2019-12-15/published/2024-07-05

[1.5in]

`covered_bond_dual_recourse`
Capped issuer, cover-pool and residual-estate recovery

Description

Allocates covered-bond payments under externally supplied legal, segregation, maturity-extension and recovery conventions.

Usage

```
covered_bond_dual_recourse(cover_pool, cover_pool_paths,
    bond_contract,
  issuer_paths, context = NULL)
```

Arguments

`cover_pool`

Bound mortgage or public-sector CreditAssetPool.

`cover_pool_paths`

Matching complete-runoff AssetPoolPaths with physical shared scenarios.

`bond_contract`

Versioned identity, currency, legal_profile_id, principal, original and optional extended maturity, dated original/extension coupons, extension rule, segregation/estate conventions and shared innovation hash.

`issuer_paths`

Scenario rows with nullable issuer_default_date, dated issuer_capacity, extension_triggered, estate_payment_date, estate_recovery_fraction and estate_capacity.

`context`

Optional compatible physical currency/conditioning and final calendar-day settlement horizon.

Details

Cover collections accumulate net of pool costs in a segregated account. Before issuer default, issuer capacity pays due claims; after default, cover cash pays. Default precedes same-day issuer payments. Interest ranks before principal. Declared extension requires issuer default by original maturity and shifts principal while activating only explicitly supplied extra coupons.

Residual estate settlement follows complete cover runoff and every possible maturity date. Recovery is capped by remaining claim and estate capacity, preventing double recovery. Original-date deferral, effective-maturity shortfall, timely-interest failure and ultimate loss remain separate. Legal enforceability and national insolvency rules are external assumptions. No penalty interest or automatic recovery uplift is inferred.

Value

CalcResult with a bound CoveredBondRecoursePaths artifact, source-separated cashflows, account reconciliation and scenario ending shortfalls.

References

https://finance.ec.europa.eu/banking/banking-regulation/covered-bonds_en

[1.5in]

credit_data
Local typed credit data and dated observation preparation

Description

Native offline CSV/JSON import, relational checks, bitemporal joins, first-default windows, work-out linkage and group-isolated chronological splits.

Usage

```
read_local_dataset(path, schema, encoding = "utf-8",
    delimiter = ",",
                   decimal_mark = ".")
validate_credit_dataset(tables, contract_version = "1.0")
join_as_of(left, right, entity_keys = "entity_id", as_of = "as_of",
           known_at = "known_at", observed_at = "observed_at",
           available_at = "available_at", release_id = "release_id",
           suffix = "_right")
build_observation_windows(ratings, default_events, exits,
    horizon_days,
                          evaluation_cutoff,
    entity_key = "entity_id")
build_recovery_episodes(default_events, recovery_cashflows,
    cures = NULL,
                        reopening_days = 90,
    evaluation_cutoff = NULL, closures = NULL)
make_grouped_temporal_splits(panel, train_end, calibration_end,
    test_end,
                             embargo_days = 0,
    entity_key = "entity_id",
                             group_key = "group_id",
    time_key = "observed_at",
                             outcome_end_key = "outcome_end")
```

Arguments

path

Local CSV or JSON path; URIs and UNC paths are rejected.

schema

Named list with version "1.0", named columns, optional primary_key and allow_empty. Each column declares type string/integer/number/boolean/date, optional nullable, and unit for numeric types.

`encoding`, `delimiter`, `decimal_mark`

Explicit text encoding, CSV separator, and decimal mark (point or comma). No grouping separators.

`tables`

Named lists or data.frames: obligors, observations, facilities, defaults and recoveries; required parent tables must be present.

`contract_version`

Supported relational contract version, `"1.0"`.

`left`, `right`

Lists of row lists or data.frames with keys and date columns.

`entity_keys`

One or more matching entity columns.

`as_of`, `known_at`, `observed_at`, `available_at`, `release_id`, `suffix`

Column names and right-column collision suffix. Release IDs must be unique.

`ratings`

Rows with observation_id, entity_id, as_of, optional performance_start, followup_end, funded and outcome_available_at (complete outcome ascertainment availability date).

`default_events`

For observation windows: event_id, entity_id, default_at, known_at. For recovery linkage: episode_id, entity_id, default_at, known_at, ead and currency.

`exits`

Rows with entity_id, exit_at and known_at. Empty list is allowed.

`horizon_days`

Positive integer performance horizon in calendar days.

`evaluation_cutoff`

ISO knowledge cutoff date. NULL for recovery means all provided history.

`entity_key`, `group_key`, `time_key`, `outcome_end_key`

Column names for identity, optional connected group, observation time and complete outcome-window end.

`recovery_cashflows`

Rows with cashflow_id, episode_id, cashflow_at, known_at, amount, kind (recovery/cost), currency. Positive costs are subtracted.

`cures`

Optional rows with episode_id, cure_at, known_at, optional closed_at with required closed_known_at on or after closure.

`closures`

Optional legal terminal rows with unique closure_id and episode_id, closed_at, known_at on or after closure, and closure_reason SETTLEMENT, WRITE_OFF or LEGAL_TERMINATION. These close a workout without asserting borrower cure. Only known terminals apply. Conflicting terminals for a merged workout, including legacy cure closure, fail; closure cannot precede any default in that workout. Post-closure cashflows fail. Output retains closure_id and closure_reason.

`reopening_days`

Nonnegative integer days after cure during which re-default belongs to the same workout; an uncured workout is also continued.

`panel`

Rows with unique observation_id, string entity ID, observation and outcome-end dates; optional string group ID.

`train_end`, `calibration_end`, `test_end`

Strictly increasing ISO end dates.

`embargo_days`

Nonnegative calendar days around split boundaries.

Details

Imports require exact columns, finite numeric values, explicit numeric units, true/false boolean literals, and JSON-safe integers. Integer text contains signed digits only; IDs larger than 9007199254740991 are strings. Empty files require allow_empty. Duplicate JSON object keys are rejected. Null or empty text is missing only in nullable columns.

Relational tables have keys obligors(entity_id), observations(entity_id, observed_at), facilities(facility_id), defaults(episode_id), recoveries(cashflow_id). Observations require available_at; facilities require entity_id and nonnegative ead; defaults require entity_id and default_at; recoveries require episode_id, cashflow_at, nonnegative amount and kind. Entity foreign keys reference obligors; recoveries reference defaults. Duplicate/missing keys, invalid dates, negative amounts and recovery before default are reported rather than dropped.

The as-of join selects latest observed_at not after as_of, then latest available_at not after known_at. Same-date ties raise DATA_AMBIGUOUS_JOIN. Unmatched rows remain with NULL right fields. This is a fact/revision join, not an interval join.

First-default windows use (origin, origin+horizon] and known events only. Prior default excludes a row. An observed in-window event yields one, complete follow-up without default yields zero, and censored or unfunded rows have missing targets. A future outcome_available_at produces outcome_pending and a missing target unless default is already observed; calendar maturity alone is insufficient. Default wins a same-day exit tie. A rejected application never becomes a good loan label. Complete event reporting through followup_end is a caller assumption; reporting gaps require shortening follow-up. Cure/re-default targets are separate.

Workout reopening uses the preceding default's known cure date. Original EAD is not added across repeats. Unique cashflows remain attached to dated defaults and exactly one workout, with calendar days since first default. Currencies must match. Explicit closure with known closed_known_at alone closes a workout; cure and data-end do not. A cashflow after closure fails pending revised closure or explicit reopening. Amounts are undiscounted, and no LGD or discount rate is inferred.

Temporal splits connect all entity/group links transitively and reserve every component for its latest partition, including future rows beyond test end. Earlier rows are excluded. Training/calibration outcomes plus embargo end by their upper boundary; calibration/test observations are strictly after the lower boundary plus embargo. Test outcomes end by test_end. Empty partitions are reported. This evaluates unseen connected entities, not repeat-customer backtests. All routines are deterministic in-memory preparation, not empirical certification.

Value

A CalcResult with a list of named rows in `$value`, or a named report. All dates are ISO YYYY-MM-DD strings. Missing outputs remain NULL. Import provenance includes SHA-256 of original bytes. Validation returns valid, violations (one-based row numbers) and row_counts. Recovery returns episodes and cashflows. Splitting returns assignments, counts and group_overlap.

Examples

```
x <- list(list(entity_id = "A", as_of = "2024-02-01", known_at = "2024-02-01"))
y <- list(list(entity_id = "A", observed_at = "2024-01-01",
               available_at = "2024-01-05", release_id = "v1", score = 2))
join_as_of(x, y)$value
```

[1.5in]

`credit_dependence`
Explicit Dependence Models and Addressed Innovations

Description

Validate dependence matrices, fit Gaussian, t, positive Clayton and Gumbel copulas, transform shared draws, calculate default intersections and write portable innovation files.

Usage

```
validate_dependence_matrix(matrix, variable_ids, matrix_kind,
  repair_policy = "none", tolerance = 1e-10, context = NULL)
fit_copula(observations, variable_ids, family, input_kind,
    tie_policy,
  correlation_repair = "none", df = NULL, df_bounds = c(1, 100),
  target, training, model_id, model_version, context = NULL)
simulate_copula(dependence_model, independent_uniforms,
    variable_ids,
  auxiliary_uniforms = NULL, context = NULL)
joint_default_probability(p_a, p_b, event_order,
```

```
  dependence_model = NULL, joint_probability = NULL, context = NULL)
make_shared_innovations(scenario_ids, time_grid, factor_ids,
    stream_spec,
  seed, output_directory, chunk_rows = 1024, context = NULL)
```

Arguments

`matrix`

Finite square matrix in the declared variable order.

`variable_ids`

Distinct ordered variable names.

`matrix_kind`

correlation or covariance.

`repair_policy`, `correlation_repair`

none preserves the input; spectral explicitly projects eigenvalues and rescales correlations.

`tolerance`

Relative matrix validation tolerance in (0,1).

`observations`

Aligned numeric training matrix, at least three rows and two columns.

`family`

gaussian, t, clayton or gumbel.

`input_kind`

observations for average ranks/(N+1), or uniforms already strictly inside (0,1).

`tie_policy`

error rejects ties; average explicitly labels a continuous approximation.

`df`

Optional fixed positive t degrees of freedom. NULL estimates it conditionally on rank correlation.

`df_bounds`

Positive increasing bounds for t profile likelihood. Boundary estimates are explicitly marked.

`target`, `training`

Model target and training provenance, including fingerprint, split ID and knowledge date.

`model_id`, `model_version`

Frozen model identifiers.

`dependence_model`

A copula FitResult or frozen model artifact.

`independent_uniforms`

Open-uniform matrix, one column per frozen variable.

`auxiliary_uniforms`

One uniform column for t/positive Clayton, two for nonindependent Gumbel; absent otherwise.

`p_a`, `p_b`

Scalar default probabilities.

`event_order`

Two event IDs in the bivariate frozen model order.

`joint_probability`

Explicit intersection; supply this or a model, not both.

`scenario_ids`, `factor_ids`

Unique address components; reordering scenarios preserves addressed values.

`time_grid`

Nonnegative strictly increasing numeric times.

`stream_spec`

Lists with stream_id and distribution uniform or normal.

`seed`

Nonempty string included in every SHA256 address.

`output_directory`

Absent or empty local directory for float64 little-endian files and manifest.

`chunk_rows`

Positive processing batch size; cannot alter addressed draws.

`context`

Optional calculation context.

Details

Fitting uses Kendall tau inversion. The t family additionally profiles the joint copula likelihood over degrees of freedom with rank correlation fixed. Clayton and Gumbel fits are exchangeable and support nonnegative average tau. Pairwise composite likelihood diagnostics do not imply joint maximum likelihood or an uncertainty estimate. Shared frailty and scale draws are explicit. Defaults are lower-tail uniform events; latent correlation is not default correlation, and zero t correlation

does not imply independence. Matrix repairs are disclosed and are not claimed nearest-correlation solutions. Addressed uniforms use the first32 SHA256 bits at cell midpoints; normal draws use two lanes and Box-Muller cosine. Uniform files match bitwise across languages; normal transforms may differ by floating-point rounding.

Value

CalcResult for matrix validation, simulated uniforms, scalar intersection or innovation manifest; FitResult with a declarative frozen copula model for estimation.

References

Yan (2007), Enjoy the Joy of Copulas, Journal of Statistical Software 21(4), doi:10.18637/jss.v021.i04. Native copula package fitCopula documentation.

[1.5in]

`credit_links`
Dated relationships and explicit risk adjustment contracts

Description

Granular native methods for typed relationship graphs, conditional support, shared guarantee caps, rating constraints, local override audits and static financial clearing.

Usage

```
build_entity_graph(entities, edges, relationship_policy, as_of,
    known_at,
  context = NULL)
resolve_parent_links(entity_graph, control_policy, context = NULL)
assign_default_groups(entity_graph, default_link_policy,
    facility_mappings,
  context = NULL)
apply_parent_support(standalone_risk, parent_risk, support_contract,
  joint_default_probability, context = NULL)
allocate_guarantee_coverage(guarantees, claims, allocation_policy,
    as_of,
  known_at, currency, context = NULL)
apply_guarantee_risk(borrower_risk, guarantor_risk,
    coverage_allocation,
  claim_id, loss_contract, joint_default_probability,
    context = NULL)
apply_rating_constraints(base_result, rating_scale, constraints,
  representative_pd_policy, context = NULL)
apply_override(base_result, override_request, policy, as_of,
    known_at,
  context = NULL)
```

```
solve_contagion_network(liability_matrix, external_assets,
    external_liabilities,
  entity_ids, currency, model_assumptions, asset_scenarios = NULL,
  tolerance = 1e-8, max_iterations = 10000, context = NULL)
```

Arguments

`entities`, `edges`

Record lists with unique entity/edge IDs. Edges contain source/target, type, validity dates, known-at and evidence reference; ownership also has a fraction.

`relationship_policy`

Versioned policy with permitted edge types.

`as_of`, `known_at`

ISO observation and information-cutoff dates.

`context`

Optional compatible calculation context.

`entity_graph`

Content-bound graph result or its value.

`control_policy`

Versioned control edge types, ownership threshold and inclusive flag.

`default_link_policy`

Versioned edge types, mutual-only or explicit-undirected direction, and Boolean transitivity. Non-transitive overlapping non-cliques remain unresolved.

`facility_mappings`

Records with facility ID and entity ID.

`standalone_risk`, `parent_risk`, `borrower_risk`, `guarantor_risk`

Risk records with entity ID, PD, stage, explicit default target and applied effect IDs.

`support_contract`

Effect, assumption reference, currency, total/reserved/allocated capacity, required rescue amount and explicit parent-alive/joint-default state terms.

`joint_default_probability`

Explicit event intersection within Frechet bounds.

`guarantees`

Nominal non-reinstating cap, used cap, covered claims, coverage fraction, guarantor, currency, validity, knowledge, legal-effectiveness and evidence records.

`claims`

Known claim ID, entity ID, amount, currency and future due date records.

`allocation_policy`

Versioned complete guarantee/claim priority orders and `cap_convention='non_reinstating_total'`.

`currency`

Reporting currency shared by all amounts; no implicit FX conversion.

`coverage_allocation`

Sealed allocated guarantee artifact.

`claim_id`

Claim belonging to the supplied borrower.

`loss_contract`

New effect ID, currency, assumption reference, `stage='PRE_GUARANTEE'`, valuation date equal to claim due date, and borrower-only/joint-default losses, payout fractions and evidence.

`base_result`

PD/grade record for constraints; named original or sealed prior artifact for overrides.

`rating_scale`

Versioned validated scale ordered by increasing risk.

`constraints`

Ordered unique rules with evidence: grade bounds (best/worst allowed) or notch bounds (reference grade, maximum improvement/deterioration).

`representative_pd_policy`

`preserve_pd` or `representative_on_binding`.

`override_request`

ID, target, value, reason, author, approver, recorded-at and validity.

`policy`

Versioned override target paths with probability, allowed-grade or bounded numeric domains and Boolean dual-control requirement.

`liability_matrix`

Nonnegative matrix with debtor rows, creditor columns and zero diagonal.

`external_assets`, `external_liabilities`

Aligned nonnegative external amounts.

`entity_ids`

Distinct IDs in matrix order.

`model_assumptions`

Exactly: model eisenberg_noe, seniority single_pari_passu, bankruptcy_costs none, outside_creditor sink.

`asset_scenarios`

Optional unique scenario IDs and alternative external asset vectors.

`tolerance`

Node-relative residual/default tolerance between 1e-12 and 1e-4.

`max_iterations`

Monotone lower-iterate cap, integer between 1 and 1000000.

Details

Graph intervals are left closed and right open. Parent ambiguity and cycles remain unresolved; ownership never establishes common PD or support by itself. Support integrates conditional rescue in parent-alive and joint-default states, requiring funding sufficient for the whole stated rescue need. Its output is before transfer.

Guarantee allocation consumes each shared cap once across the supplied claims. Payment protection preserves borrower PD and reduces conditional monetary loss by min(covered amount, positive loss) times supplied conditional payout fraction. Previously guarantee-adjusted inputs are rejected even under a new effect ID. Several guarantors require a joint event model. Nominal cap and loss use the claim due-date valuation basis; discount state losses separately.

Rating rules are intersected and inconsistent bounds fail. Nonbinding constraints preserve continuous PD exactly. Grade-only policy exposes any PD/grade inconsistency; representative policy changes PD only on binding. Override history preserves original values and content hashes, rejects duplicate IDs, and records expired requests without applying. A changed evaluation cutoff requires replay from the original; copying an overridden current value is not an unadjusted baseline. Only the requested field changes.

Clearing uses limited liability, absolute priority and one pro-rata debt class. Native lpSolve maximizes payment fractions to select greatest Eisenberg–Noe clearing. A monotone lower iteration and greatest/lower gap diagnose numerical enclosure or multiple clearing points; a small iterate change alone does not prove uniqueness. Outside creditors do not recycle payments. Extreme amount dynamic ranges below 1e-10 of the largest amount fail explicitly. This is a static financial network, not a group PD estimator, dynamic insolvency model or multi-seniority waterfall.

Value

A CalcResult containing the graph, mapping, adjusted risk, immutable local artifact or clearing payments and sensitivity, with diagnostics and limitations.

References

Eisenberg, L. and Noe, T. H. (2001). Systemic Risk in Financial Systems. Management Science 47(2), 236–249. doi:10.1287/mnsc.47.2.236.9835.

Examples

```
a <- list(model="eisenberg_noe", seniority="single_pari_passu",
          bankruptcy_costs="none", outside_creditor="sink")
solve_contagion_network(matrix(c(0,50,50,0),2), c(20,10), c(50,50),
  c("A","B"), "EUR", a)$value$base$clearing_payments
```

[1.5in]

`credit_loss_distribution`
Portfolio Loss Distribution, Tail Allocation and Concentration

Description

Weighted discrete loss quantiles and exact upper-mass expected shortfall, paired scenario changes, concentration and a homogeneous Gaussian factor benchmark.

Usage

```
credit_loss_distribution(loss_paths, weights, scenario_ids,
  levels = c(.95, .99), currency, context = NULL)
allocate_expected_shortfall(position_loss_paths, weights,
    scenario_ids,
  position_ids, level, currency, groups = NULL, context = NULL)
incremental_credit_risk(base_loss_paths, change_loss_paths, weights,
  scenario_ids, levels = c(.95, .99), currency, change_id,
    context = NULL)
credit_concentration_measures(positions, weight_field, group_fields,
  weight_unit, context = NULL)
asymptotic_factor_loss(pd, lgd, asset_correlation,
    systematic_factor,
  context = NULL)
```

Arguments

`loss_paths`, `base_loss_paths`, `change_loss_paths`

Finite signed monetary paths; changes are paired with base paths on identical scenarios.

`weights`

Aligned nonnegative scenario weights with positive total; normalized internally.

`scenario_ids`, `position_ids`

Distinct nonempty identifiers in input order.

`levels`, `level`

Tail probability levels in [0,1]; level one uses the maximum-loss limit.

`currency`

Explicit reporting currency matching context.

`context`

Optional calculation context.

`position_loss_paths`

Scenario by position monetary loss matrix.

`groups`

Optional named list assigning exactly one group to each position.

`change_id`

Explicit identifier for the paired portfolio change.

`positions`

Records with unique position_id, nonnegative declared weight and group fields.

`weight_field`, `group_fields`, `weight_unit`

Weight column, grouping columns and explicit exposure/risk unit.

`pd`, `lgd`, `asset_correlation`

Homogeneous scalar unconditional physical PD, fixed raw LGD and latent asset correlation in [0,1).

`systematic_factor`

Finite standard-normal factor values; negative is adverse.

Details

VaR uses the left inverse CDF. ES selects exactly the upper 1-alpha mass, allocating partial boundary atoms proportionally; level zero is the mean and level one the maximum-loss limit. Signed hedge allocations reconcile to ES. These are supplied-distribution calculations with no inferred confidence intervals. HHI and entropy are descriptive, not regulatory limits. The factor benchmark assumes infinitely granular identical exposures; latent asset correlation is not default correlation. Its unconditional factor law does not establish empirical TTC calibration.

Value

CalcResult with the requested distribution, allocations, paired changes, concentration dimensions or conditional Gaussian benchmark.

References

Rockafellar and Uryasev, Conditional Value-at-Risk for General Loss Distributions. https://sites.math.washington.edu/~rtr/papers/rtr187-CVaR2.pdf

[1.5in]

`credit_permutation_importance`
Frozen Held-Out Grouped Permutation Importance

Description

Compare paired prediction losses under explicit shared permutation plans, without refitting or selecting features.

Usage

```
credit_permutation_importance(predictor, X, y, model_contract,
  feature_groups, permutations, design, loss_contract, row_metadata,
  weights = NULL, strata = NULL, custom_loss = NULL,
  support_evaluator = NULL, support_contract = NULL, context = NULL)
```

Arguments

`predictor`

Frozen local prediction function accepting a row matrix.

`X`

Held-out feature matrix or row lists.

`y`

Numeric held-out mature labels.

`model_contract`

Frozen identity, feature order, output scale and prediction target.

`feature_groups`

Named nonoverlapping partition of all features.

`permutations`

Matrix or row lists of complete zero-based permutation bijections. Each plan is shared across blocks.

`design`

split OUT_OF_TIME/HOLDOUT, training_end, data_cutoff, train_row_ids, train_subject_ids, model_and_calibration_frozen TRUE, selection_use FORBIDDEN and permutation_design FIXED_PERMUTATIONS/IID_RANDOM_PERMUTATIONS.

`loss_contract`

loss_id, version and kind BRIER, LOG_LOSS, MSE, MAE or CUSTOM. Binary probability losses require binary outcomes and probability outputs.

`row_metadata`

One row_id, subject_id, observation_time, target_end and label_known_time record per evaluation row.

`weights`

Optional nonnegative row weights with positive total; normalized internally.

`strata`

Optional row labels; every permutation must preserve each recipient stratum.

`custom_loss`

For CUSTOM only, function(y, predictions) returning one finite loss per row.

`support_evaluator`

Optional local function from row matrix to one boolean support flag per row.

`support_contract`

support_id and version, required with support_evaluator.

`context`

Optional common calculation context.

Details

Blocks are permuted jointly; outcomes and weights remain with recipients. Fitting-identity overlap, incomplete targets, future-known labels and invalid permutation plans are rejected. Positive-weight infinite log loss is not clipped and raises an explicit non-estimability error. Unsupported rows are reported and retained in denominators. Weighted feature correlation is descriptive; cross-block dependence can be broken. Only declared IID random permutations with at least two plans receive a standard error of the mean loss increase. This excludes cohort, model and selection uncertainty. Fixed plans receive none. No refitting or feature selection occurs; supplied provenance is checked but not authenticated.

Value

CalcResult with baseline and per-plan losses, grouped paired increases, permutation uncertainty, support findings, correlations and content hashes.

[1.5in]

`cumulative_pd_from_conditional`
First-Default Probabilities from Period Hazards

Description

Converts conditional probabilities into cumulative first-default probabilities or unconditional period first-default masses.

Usage

```
cumulative_pd_from_conditional(conditional_pd, context = NULL)
marginal_pd_from_conditional(conditional_pd, context = NULL)
```

Arguments

`conditional_pd`

A finite numeric vector of chronological first-default probabilities, each conditional on survival to the period start. Values must lie between zero and one.

`context`

Optional calculation context.

Details

Survival through period k is the product of one minus each preceding hazard, including the current hazard. Cumulative PD is one minus survival. Period default mass is the current hazard times survival to its start. Calculations use log-survival for numerical accuracy; marginal masses are computed directly rather than by differencing cumulative probabilities. A unit hazard absorbs all remaining survival. Empty inputs return empty vectors. No cure, repeated default, interpolation or annualization is implied.

Value

A `CalcResult` whose `value` is a probability vector in chronological order.

Examples

```
cumulative_pd_from_conditional(c(0.1, 0.2))$value
marginal_pd_from_conditional(c(0.1, 0.2))$value
```

[1.5in]

`decision_policy_backtest`
Historical Decision-Policy Backtest

Description

Audit frozen policy decisions, actual offers and funding stages, matched booked outcomes, declared scenario constraints and paired policy sensitivity.

Usage

```
decision_policy_backtest(evaluated_grid, grid_inputs,
    historical_cases,
  design, reference_candidate_id, uncertainty = "none",
  cluster_resamples = NULL, confidence_level = .95,
  tolerance = 1e-10, context = NULL)
```

Arguments

`evaluated_grid`

Sealed DecisionGrid from evaluate_decision_grid.

`grid_inputs`

Exact original named arguments to evaluate_decision_grid, without context.

`historical_cases`

Dated validation records with case and historical candidate identifiers, sealed historical decision, actual offer/take-up stages and their knowledge date, nullable offered limit and booked principal, optional known complete application costs.

`design`

Physical target, currency, split, data fingerprint and cutoff matching the frozen grid.

`reference_candidate_id`

Reference policy for paired scenario differences.

`uncertainty`

None or cluster_bootstrap.

`cluster_resamples`

Supplied whole-cluster draws.

`confidence_level`

Interior percentile interval level.

`tolerance`

Nonnegative absolute/relative numerical reconciliation tolerance.

`context`

Optional compatible calculation context.

Details

Evaluation profile target_contract_version 1.1 requires a complete target with performance_origin first_draw_at and event default or first_default. The validation design repeats policy_target_contract_version 1.1 and performance_origin, preserving the exact original target. Missing version or 1.0 remains default-only; unknown versions are rejected. Observed decision stages do not imply observed economic outcomes. No lending action or fit is executed. Actual funding requires take-up and an offer. Whole offer hashes bind terms, so changed price/limit scenarios do not inherit factual credit labels. Unfunded outcomes are unknown. Offered limits require frozen offer contract_terms.limit. Mature matched-funded risk is separate from assumed offer economics. Schema 1.1 approval applicability and additional gates are replayed by the real evaluator. Declared scenario violations and historical inconsistencies block reconciliation. Agreement alone does not approve a policy. Full-policy causal value remains unidentified; this validator does not assert an off-policy identification design.

Value

CalcResult with ValidationResult, factual and scenario metrics, paired contrasts and cluster uncertainty, historical audits, constraints and counterfactual applicability.

References

Ehrhardt et al., Reject Inference Methods in Credit Scoring. https://adimajo.github.io/assets/publications/rejectInference.pdf

---

[1.5in]

`decisioning_offer_economics`
Local Limit, Quote and Treatment Scenario Economics

Description

Enumerates supplied local offers and action-conditioned cashflows with explicit risk, affordability and eligibility assumptions. No decision is executed and no causal price or treatment effect is estimated.

Usage

```
compute_limit_frontier(offer_candidates, constraint_profile,
  as_of, context = NULL)
compute_risk_based_price_grid(quote_candidates, pricing_profile,
  as_of, context = NULL)
evaluate_treatment_candidates(treatment_candidates,
    treatment_profile,
  as_of, context = NULL)
```

Arguments

`offer_candidates`

Records with offer_id, version, known_at, currency, subject_id, amount, integer tenor_days, assumption_reference, target, pd, lgd, ead, risk_conditioning=CONDITIONAL_ON_THIS_OFFER, risk_reference, affordability_status, explicit nullable affordability_maximum_amount and eligibility_status. One subject and distinct offer IDs required.

`constraint_profile`

Versioned profile with profile_id, version, known_at, currency, target, maximum_amount, maximum_ead, maximum_expected_loss, maximum_concentration, positive concentration_denominator, existing_group_exposure and nonnegative tolerance.

`quote_candidates`

Versioned offer records as above, annual_rate, target, risk_conditioning=CONDITIONAL_ON_THIS_ACCEPTED_QUOTE, risk_reference, takeup_probability, takeup_reference, upfront_fee, origination_cost, offer_cost and periods. Each period specifies payment_date, scheduled_principal,

conditional_default_probability, conditional_prepayment_probability, lgd, recovery_date, payment_discount_factor, recovery_discount_factor, servicing_cost and prepayment_fee.

`pricing_profile`

Versioned common profile plus default_timing=BEFORE_DUE, prepayment_timing=AFTER_COUPON, funding_treatment=DISCOUNT_CURVE_ONLY, selection_assumption=QUOTE_CONDITIONED_INPUTS_NOT_CAUSAL.

`treatment_candidates`

Records with treatment_id, version, known_at, subject_id, currency, action, prediction_reference, support_status, affordability_status, eligibility_status, uptake_probability, offer_cost, accepted_upfront_cost, rejected_present_value and cashflows. Each cashflow has cashflow_id, date, expected_recovery, expected_cost and discount_factor.

`treatment_profile`

Versioned common profile plus subject_id, baseline_id, baseline_present_value, identification=CONDITIONAL_SCENARIO_NOT_CAUSAL and allowed_actions.

`as_of`

ISO decision and present-value anchor date. All profiles and candidate assumptions must already be known.

`context`

Optional calculation context, compatible currency and physical probability measure.

Details

Targets specify physical default, positive integer horizon, time_unit=days and explicit conditioning. Limit candidates share one comparison target, although tenors may differ. Price targets span the entire quote tenor and end on its last payment. Offer-conditioned risk is supplied, never copied implicitly from a baseline borrower PD.

Limit EL is PD times conditional economic LGD times EAD. Concentration is existing group exposure plus candidate EAD divided by the declared fixed post-decision denominator. Numerical tolerance applies in the units of each comparison. Slacks, violations and binding flags remain separate from affordability and eligibility gates. Unknown affordability caps are infeasible. Feasible offers are enumerated without monotone-risk assumptions. Same-tenor domination means at least as much amount, no more EL and one strict improvement. No hidden cross-tenor utility or interpolation is used.

Pricing starts survival at one. Default and prepayment probabilities are mutually exclusive conditional on opening survival; masses are survival times their respective hazards. Surviving principal follows the supplied full amortization schedule. Fixed simple ACT/365F coupons use opening principal. Default before due misses coupon and exposes opening principal; recovery is principal times one minus LGD at the supplied later date. LGD describes undiscounted net recovery at that date, and may be signed. An already discounted LGD would double discount. Prepayment after coupon returns all opening principal plus its fee. Continuing accounts pay coupon and scheduled principal. Servicing cost applies to all opening survivors. Full surviving repayment is required; no residual value is assumed. Same-date funding discount factors must agree within a quote.

Booked NPV subtracts initial amount and origination cost, adds upfront fee and discounted expected cashflows. Funding enters only through discount factors. Reported EL is already reflected in recovery cashflows and must not be subtracted again. Per-offer NPV is takeup times booked NPV minus offer cost, paid once regardless of uptake. No demand elasticity or unique break-even rate is inferred.

Treatment expected recovery/cost cashflows are already event-probability-weighted within acceptance. Accepted value subtracts accepted upfront cost; total expected value mixes uptake and explicit rejected value and subtracts offer cost once. The baseline is an independent comparator. Unsupported predictions give null values and cannot be ranked. Supported but unaffordable treatments retain their value and infeasibility reason. Allowed supported feasible treatments rank by expected value, ties by ID. Historical selection and missing counterfactuals are not repaired by this conditional scenario comparison.

Value

CalcResult with SHA-256-bound LimitFrontier, PriceGrid or TreatmentEvaluation, candidate fingerprints, monetary components, separate feasibility reasons and declared assumptions.

[1.5in]

`decisioning_policy_evaluation`
Held-Out Policy Evaluation, Constrained Selection and Robustness

---

Description

Replays frozen local policies, separates supplied prospective offer economics from observed matched-funded evidence, and checks finite shared perturbations without inferring causal policy effects.

Usage

```
evaluate_decision_grid(policy_candidates, cases, candidate_inputs,
  observed_outcomes, evaluation_profile, evaluation_cutoff,
    context = NULL)
select_constrained_policy(evaluated_grid, context = NULL)
assess_policy_robustness(policy, baseline_cases,
    perturbation_scenarios,
  robustness_profile, as_of, context = NULL)
```

Arguments

`policy_candidates`

Records candidate_id, raw frozen DecisionPolicy value in policy, and offer_rule_reference. Each complete candidate must match its predeclared content hash.

`cases`

Records case_id, decision_date, nonnegative weight, split_id, prediction_reference, excluded_from_model_training=TRUE and excluded_from_calibration=TRUE. Total weight must be positive.

`candidate_inputs`

Complete candidate/case grid with candidate_id, case_id, separate dated risk, affordability, eligibility and operations objects, explicit nullable offer, and nonapproval_costs for REFER, DECLINE and BLOCKED.

`observed_outcomes`

Records case_id, funded, complete, and explicit nullable booked_offer_hash, funded_at, funding_known_at, outcome_end, known_at, defaulted, economic_loss and net_profit. Funding and outcome knowledge are separate.

`evaluation_profile`

Versioned profile_id, version, known_at, currency, registered_at, outcome_access_date, evaluation_role, prediction_design, split_id, data_fingerprint, target, candidate_hashes, objective, constraints, tie_break, tolerance, loss_bounds and profit_bounds. See Details.

`evaluation_cutoff`

ISO economic evidence cutoff.

`evaluated_grid`

Bound DecisionGrid value or CalcResult.

`policy`

Frozen DecisionPolicy value or CalcResult.

`baseline_cases`

Records case_id, decision_date, weight, four source objects, nullable offer_terms, utility_reference, utility_by_action and feasible_by_action.

`perturbation_scenarios`

Records scenario_id, weight, perturbation_reference and complete cases tables preserving baseline IDs, dates and case weights.

`robustness_profile`

Versioned common profile plus target, scenario_design (BOUNDED_GRID, BOOTSTRAP_REPLAY or STRESS_SET), generation_reference, interpretation=PERTURBATION_NOT_CAUSAL and tolerance.

`as_of`

ISO robustness evaluation date.

`context`

Optional compatible physical target/currency calculation context.

Details

Evaluation profile target_contract_version 1.1 requires a complete target with performance_origin first_draw_at and event default or first_default. The validation design repeats policy_target_contract_version 1.1 and performance_origin, preserving the exact original target. Missing version or 1.0 remains default-only; unknown versions are rejected. Observed decision stages do not imply observed economic outcomes.

Evaluation roles are VALIDATION, OOF_VALIDATION or TEST; prediction design is HELD_OUT or OUT_OF_FOLD. OOF_VALIDATION requires OUT_OF_FOLD. Per-case model and calibration exclusion and pre-registration are caller assertions, not an independent audit. Register candidates, objective and constraints before declared outcome access. Policies use the same physical default target with integer day horizon. Historical policy knowledge cutoff equals the case decision date.

Approved cases need versioned offer_id, version, known_at, currency, target, risk_conditioning=CONDITIONAL_ON_THIS_ACCEPTED_OFFER, assumption_reference, support_status, and nullable takeup_probability, conditional_booked_expected_loss, conditional_booked_npv, ead and offer_cost. SUPPORTED assumptions require finite values; positive-weight unsupported approved offers make monetary metrics null. Per-applicant NPV is takeup times book NPV minus offer cost; book NPV already includes losses and funding. EL is a separate metric, not subtracted again. Nonapproval costs are explicit. All metrics use the same total applicant weight.

Unfunded rows cannot supply fabricated defaults, losses or profits. Funded outcome horizons begin at actual funding and equal the target horizon. Funding knowledge cannot precede funding. Complete labels need outcome knowledge no earlier than horizon end; incomplete labels remain null. Observed results require recommended approval, exact booked-offer hash match, funding known by cutoff and, for mature statistics, complete outcomes known by cutoff. Changed assumptions or versions are deliberately not treated as equivalent offers. Missing funding, unfunded, different offers and incomplete outcomes have distinct statuses. Rejected cases are never counted as nondefaults. Loss/profit bounds cover only missing outcomes inside the known matched-funded cohort, not unknown funding or causal policy value. Full candidate causal value remains null.

Selection accepts only VALIDATION/OOF_VALIDATION. TEST grids may be evaluated but cannot select policy. The objective is expected_npv_per_application and tie break candidate_id. Each constraint has distinct constraint_id, metric, operator (le/ge) and threshold. Metrics include approval_rate, refer_rate, decline_rate, blocked_rate, expected_loss_per_application, expected_npv_per_application and expected_funded_ead_per_application. Slacks and binding flags are recomputed. Unidentified candidates are infeasible. Objective/constraint-direction Pareto domination is checked before utility/ID ties. Empty feasible sets return INFEASIBLE, not an arbitrary candidate. Full candidate and policy bindings are verified; the selected policy is returned unchanged. Final assessment requires an untouched test period.

Robustness replays one fixed policy on a finite shared scenario set. Scenario weights are nonnegative and sum to one. Each case utility/feasibility map explicitly covers APPROVE, REFER, DECLINE and BLOCKED, with nullable finite monetary utility and nullable Boolean feasibility. Approved cases require offer_id, version, amount and annual_rate terms. These supplied terms are not computed by the rule policy. Action flips, changed input sources, both-approved amount/rate changes, feasibility and regret remain separate. A changed source is associated with a flip, not causal attribution.

Regret compares only explicitly feasible actions with known utilities. Unknown positive-weight case/scenario comparisons propagate null worst-case regret. Known infeasible action rates are ac-

companied by unknown-dimension findings. Worst-case mean regret ranges only over supplied positive-weight scenarios. Bootstrap validity and dependence are caller responsibilities; no automatic confidence interval, counterfactual identification or continuous uncertainty bound is asserted. No financial action is executed.

Value

CalcResult with bound DecisionGrid, PolicySelection or PolicyRobustness, input and artifact hashes, complete rule traces, separate assumption/outcome metrics, feasibility reports and explicit unidentified quantities.

References

https://scikit-learn.org/stable/modules/classification_threshold.html separates probability fitting and threshold selection and warns against tuning on the same training observations; reviewed 2026-10-13. Local contracts and economic algorithms here are library-specific.

[1.5in]

`detect_cashflow_default`
Cashflow Double Test and Independent Default Triggers

Description

Evaluate simultaneous debt-service and value-coverage breaches, independent unlikeliness-to-pay and material-arrears triggers, and absorbing first default.

Usage

```
detect_cashflow_default(path_records, default_policy, currency,
    context = NULL)
```

Arguments

`path_records`

Record lists containing scenario_id, entity_id, facility_id, date, currency, cfads, debt_service_due, debt_outstanding, asset_value, materiality_met, material_days_past_due, liquidity_buffer_available and utp_indicators. Monetary values use one reporting currency. CFADS and debt service cover the same period; debt and asset values share a date and valuation perimeter. Empty UTP indicators represent an assessed absence of other signals, never unknown evidence.

`default_policy`

Versioned list with policy_id, version, basis `BASEL_CRE36_REFERENCE`, assessment_level (obligor or facility), dscr_threshold, ltv_threshold, dscr_comparison (lt/le), ltv_comparison (gt/ge), past_due_days_threshold, past_due_comparison (gt), past_due_rule_reference, materiality_rule_reference, utp_rule_reference, logical dual_test_as_utp, economic_joint_timing (SAME_DATE),

default_state (ABSORBING_FIRST_DEFAULT). Non-90 day thresholds through 180 require supervisory_exception_eligibility (RETAIL_OR_PSE_APPROVED); facility scope requires facility_scope_eligibility (RETAIL_APPROVED). These declarations do not establish legal eligibility.

`currency`

Reporting currency string, consistent across rows and context.

`context`

Optional calculation context.

Details

The economic double test requires simultaneous DSCR and LTV breaches on the same facility and date. A declared policy may use it as a UTP indicator only while a debt-service shortfall remains after explicitly available liquidity. Buffers must be legally and operationally available by the payment date, net of prior claims; hypothetical collateral proceeds or uncommitted refinancing do not qualify. They never cancel other independently established UTP signals. Regulatory-reference default is UTP OR material arrears exceeding the applicable day threshold. Adequate value never cancels either trigger. Zero debt service has no DSCR breach; positive debt and zero assets have an LTV breach. Supplied evidence must cover the assessment subject, including relevant obligations outside the simulated facility. Material-arrears counters follow the referenced materiality rule and must advance by calendar time or reset within an interval. Obligor assessment requires a common facility grid. Default remains absorbing; cure and redefault require separate episode logic. Detection dates are grid endpoints, with the preceding nondefault endpoint reported. Initial defaults are left censored and excluded from initially performing PD risk denominators.

Value

A CalcResult containing policy and hash, facility_tests, assessments and first_default_events, including initial_default, last_nondefault_date, observed_through and default_time_status.

References

Basel Framework, CRE36.68–36.75: https://www.bis.org/committees/bcbs/basel-framework/standard/cre?allChapters=true&chapter=36.

[1.5in]

`ead`

Dated Drawn Exposure Schedules and Realized Conversion Factors

---

Description

Accrue balances through explicit dated events with known default timing, and retain raw realized CCF values with undefined-denominator statuses.

Usage

```
exposure_schedule(contract, events, annual_interest_rates,
    period_end_dates,
  day_count, default_timing, allow_overlimit, currency,
    context = NULL)
realized_ccf(reference_drawn, reference_limit, ead_at_default,
    reference_date,
  default_date, reference_known_at, default_known_at,
    evaluation_cutoff,
  currency, minimum_undrawn = 0, context = NULL)
```

Arguments

`contract`

Named list with facility_id, currency, ISO as_of, and nonnegative drawn_principal, credit_limit and accrued_interest. Snapshot balances already include all activity through as_of.

`events`

Data frame or list of rows with unique event_id, event_date, known_at, nonnegative integer sequence, kind and nonnegative amount. Kinds are DRAW, PRINCIPAL_REPAYMENT, INTEREST_PAYMENT and LIMIT_CHANGE. A limit change amount is the new total limit. All events follow the snapshot, fit the horizon and must already be known at snapshot time. Duplicate date/sequence pairs are rejected.

`annual_interest_rates`

One finite nonnegative simple annual rate per period.

`period_end_dates`

Nonempty character vector/list of increasing ISO calendar dates after the snapshot.

`day_count`

ACT/365F or ACT/360, explicitly selected for annual simple interest.

`default_timing`

BEFORE_PERIOD_END_EVENTS or AFTER_PERIOD_END_EVENTS, referring to all scheduled events on the endpoint date.

`allow_overlimit`

Explicit logical permission for new draws above the limit. Initial overlimit amounts or limit cuts remain visible separately.

`currency`

Explicit nonempty currency, matching contract and context.

`context`

Optional calculation context or context result.

`reference_drawn`, `reference_limit`, `ead_at_default`

Finite nonnegative monetary amounts under a common currency/exposure definition.

`reference_date`, `default_date`, `reference_known_at`, `default_known_at`, `evaluation_cutoff`

ISO dates satisfying reference_known_at at or before reference_date, strictly before default_date, at or before default_known_at, at or before evaluation_cutoff.

`minimum_undrawn`

Nonnegative monetary threshold; a positive denominator at or below this threshold does not produce a numerical CCF.

Details

Interest accrues on drawn principal between successive events using actual calendar days and the declared divisor. It is not capitalized. Principal and interest repayments are explicit separate events and cannot exceed the corresponding balance. The event ledger exposes every applied balance change. A bullet payment can give positive before-event exposure and zero after-event principal; interest persists until paid explicitly.

CCF is (EAD at default minus reference drawn) divided by (reference limit minus reference drawn). Negative and above-one CCF values are retained. Nonpositive undrawn commitment yields NULL CCF with NO_POSITIVE_UNDRAWN; a small positive commitment at/below the threshold yields BELOW_MINIMUM_UNDRAWN. No artificial floor, ceiling, limit-change adjustment, cancellation-right model, future borrower behavior or legal netting rule is inferred.

Value

A CalcResult. exposure_schedule returns a schedule and event ledger bound by schedule_hash, with facility/valuation conventions. Each period contains opening and closing principal, accrued interest, limit, nonnegative undrawn commitment, overlimit amount, exposures before/after endpoint events and selected conditional_ead. realized_ccf returns raw ccf, denominator_status, reference amounts, undrawn commitment, exposure increase and reconstructed EAD when defined.

Examples

```
realized_ccf(40, 100, 70, "2023-12-01", "2024-01-01",
  "2023-12-01", "2024-01-02", "2024-01-31", "EUR")$value
```

[1.5in]

`ead_counterparty`
Signed Netting, Collateral Unit Ledgers and Frozen Closeout Exposure

---

Description

Apply explicit legal membership, margin, market and timing contracts to user-valued aligned counterparty scenarios.

Usage

```
aggregate_netting_set_exposure(trade_value_paths,
    netting_agreements, fx_paths,
  as_of, reporting_currency, valuation_convention, context = NULL)
simulate_collateral_margin(netting_paths, margin_policies,
    call_calendar,
  collateral_market_paths, context = NULL)
closeout_exposure_at_default(netting_paths, collateral_paths,
    default_events,
  intermediate_cashflows, freeze_timing, cashflow_convention,
    context = NULL)
```

Arguments

`trade_value_paths`

Rows with path_id, date, trade_id, currency, signed value. A complete common scenario/date/trade grid is required.

`netting_agreements`

Rows containing netting_set_id, counterparty_id, trade_ids, legally_enforceable Boolean, evidence_reference, known_at, effective_from and explicit effective_until date or NULL. Unique complete membership is required. Multiple-trade sets require enforceability; singleton sets may declare FALSE.

`fx_paths`

Exact path_id/date/currency/rate rows. Positive reporting-currency units per trade-currency unit; explicit reporting-currency identity rate one.

`as_of`

ISO snapshot date. Agreements must be known then and effective for all valuation dates under the half-open interval [effective_from,effective_until).

`reporting_currency`

Explicit currency consistent with calculation context.

`valuation_convention`

The string EX_DIVIDEND_BEFORE_MARGIN.

`netting_paths`, `collateral_paths`

Bound CalcResult or value artifacts from aggregation and margin simulation. Hash and shared netting binding are verified.

`margin_policies`

One row per set: known_at, collateral_currency, initial_units, threshold_received, threshold_posted, minimum_transfer_amount, ownership, received_enforceable, posted_returnability, posted_claim_netting_enforceable, reuse. Ownership is TITLE_TRANSFER or SECURITY_

INTEREST; received_enforceable must be TRUE. Posted returnability is UNSECURED_CLAIM or BANKRUPTCY_REMOTE, with the latter incompatible with TITLE_TRANSFER. Posted claim netting is an explicit Boolean. Reuse is ALLOWED or PROHIBITED. Thresholds and MTA are nonnegative.

`call_calendar`

Rows with call_id, path_id, netting_set_id, valuation_date, call_date, settlement_date, dispute_delay_days. Valuation is no later than call, settlement no earlier; dispute delay adds nonnegative calendar days. One call per path/set/date. Effective settlement is on the grid or beyond its end.

`collateral_market_paths`

Complete path_id/date/netting_set_id grid with currency, positive unit_price, positive fx_rate, received_haircut and posted_haircut in [0,1). A single homogeneous asset per set is tracked in signed units.

`default_events`

Unique path/set rows containing default_date, closeout_date and margin_observation_date. Observation cannot follow default; closeout cannot precede default. Default and closeout require exact grid dates.

`intermediate_cashflows`

Unique cashflow_id rows with path/set, date, signed amount, reporting currency, and included_in_closeout_value=FALSE. All are unsettled claims excluded from clean values and dated in (default,closeout].

`freeze_timing`

BEFORE_DATE_EVENTS freezes opening collateral units; AFTER_DATE_EVENTS freezes closing units after all that day's events.

`cashflow_convention`

UNSETTLED_EXCLUDED_FROM_CLOSEOUT_VALUE.

`context`

Optional calculation context; currency must match.

Details

Signed values aggregate only within sets before positive-part flooring. No cross-set offset, interpolation or automatic legal opinion is supplied. Prior settlements precede calls, followed by immediate settlements. Target units use declared valuation-date net value, price, FX and haircut. Pending units are included in the committed balance to prevent duplicate calls. MTA is measured as gross reporting market value; future pending settlements remain visible.

Positive collateral is recognized after received haircut. Nonremote posted collateral is a full nominal return claim, while posted haircut only sizes required posting. Enforceable posted-claim netting gives max(V-received+posted,0); otherwise exposure is max(V-received,0)+posted. Reuse does not imply reinvestment benefits. Initial units are the balance before first-date events; pre-snapshot unsettled obligations require an explicitly extended scenario.

At default, frozen units are revalued at closeout market, FX and haircut; later margin is ignored. Outstanding intervening claims are included exactly once. Same-day intraday default cashflows are outside this date-only contract. Missing path extension fails. The caller establishes the margin observation anchor; regulatory MPOR floors and capital approval are not inferred.

Value

CalcResult with signed net values and positive exposures, a bound collateral scenario artifact with unit balances, pending transfers and call ledger, or closeout components and exposures. Margin-period days use the declared observation anchor; default-to-closeout days are a separate output.

References

BCBS CRE50.15–19 and 50.23, published 5 July 2024, https://www.bis.org/committees/bcbs/basel-framework/standard/cre/50/inforce/2019-12-15/published/2024-07-05.

[1.5in]

`ead_models`
Frozen CCF Conditional Means and Direct EAD Distributions

Description

Fit historically aligned conversion-factor mean regressions or nonnegative direct-exposure distributions, and replay models or contractual schedules.

Usage

```
fit_ead_model(exposure_episodes, X_train, feature_order,
    target_mode, model_spec,
  target, training, model_id, model_version, currency,
    weights = NULL,
  minimum_undrawn = 0, fit_intercept = TRUE, context = NULL)
predict_ead(fitted_model_or_schedule, facility_covariates,
    feature_order,
  expected_target, reference_dates, default_dates, currency,
  reference_drawn = NULL, reference_limit = NULL,
    quantile_levels = NULL,
  context = NULL)
```

Arguments

`exposure_episodes`

Rows with episode_id, reference_date, reference_known_at, default_date, default_known_at, features_observed_at, features_known_at, reference_drawn, reference_limit, ead_at_default, currency. Amounts nonnegative; reference and feature knowledge no later than reference date; default exactly horizon days later and known by training cutoff.

`X_train`, `facility_covariates`

Finite numeric matrices with ordered columns. For schedule prediction, facility_covariates must be NULL.

`feature_order`

Distinct predictor names; character() for no predictors.

`target_mode`

ccf or direct_ead.

`model_spec`

CCF: mean_model="identity", covariance="HC0". Direct EAD: positive_ distribution="lognormal", zero_model="none", "logit" or "all_zero".

`target`, `expected_target`

Physical given_default target, integer-day horizon, time_unit="days", event="ead" and explicit currency. Prediction requires exact match. For schedule prediction expected_target is NULL.

`training`

Metadata with data_fingerprint, split_id, known_at, optional data_origin.

`model_id`, `model_version`

Nonempty artifact identifiers.

`currency`

Explicit reporting currency, matching episodes, target and context.

`weights`

Nonnegative exact integer frequency weights, positive finite total at most 2^53-1, representing independent replicated observations.

`minimum_undrawn`

Nonnegative threshold. CCF denominators at or below it are excluded with reason during fitting and rejected during prediction.

`fit_intercept`

Whether the regression mean includes an intercept.

`fitted_model_or_schedule`

FitResult, frozen model, or bound exposure_schedule result/value. Schedule integrity is checked against schedule_hash.

`reference_dates`, `default_dates`

One ISO reference/default date per prediction. Model reference cannot precede training knowledge and horizon must match. Schedules require reference equal to as_of and exact period-end default dates.

`reference_drawn`, `reference_limit`

Aligned nonnegative reference amounts for CCF reconstruction only; NULL for direct-EAD or schedule prediction.

`quantile_levels`

Optional increasing interior probabilities, only for direct EAD distributions. These describe predictive distribution, not estimation uncertainty.

`context`

Optional calculation context, compatible with frozen target/currency.

Details

CCF is (EAD-drawn)/(limit-drawn), without clipping. Frequency least squares and HC0 covariance estimate an identity conditional mean. Reconstructed negative EAD fails rather than being floored. Mean standard errors hold reference amounts fixed; a CCF mean alone does not identify exposure quantiles.

Direct EAD models zero probability separately from positive lognormal amounts. The positive log regression uses maximum-likelihood residual variance, requires nonzero residual dispersion and full rank, and yields mean exp(mu+squared_sigma/2). Multiplication by positive probability gives total mean. Quantiles include the zero atom. none requires all positive targets; logit requires both classes; all_zero fits an exact point mass without positive-stage parameters. Positive-stage support is diagnosed separately. Predictive quantiles assume a correctly specified lognormal distribution and omit parameter-estimation uncertainty.

Predictors and knowledge dates are supplied explicitly; truth of source evidence cannot be inferred. Models use native declarative artifacts with input fingerprints, retained provenance and cross-language replay. Schedule prediction interpolates nothing and respects its before/after-payment default convention. No PD or regulatory conversion factor is supplied.

Value

FitResult with chronological diagnostics, excluded rows, regressions, optional positive-EAD gate, covariance, feature support and provenance. CalcResult predictions contain conditional_ead, dates and currency; CCF adds conditional_ccf and coefficient-based mean standard errors; direct EAD adds positive_probability and optional quantiles.

[1.5in]

`estimate_long_run_pd`
Estimate observed long-run annual default rates

Description

Aggregates explicitly selected complete mature annual cohorts available at the knowledge date.

Usage

```
estimate_long_run_pd(cohorts, selected_cohort_ids, knowledge_date,
    weighting,
  uncertainty, dependence_assumption, representativeness, target,
  confidence_level = .95, context = NULL)
```

Arguments

`cohorts`

List of records or data frame: cohort_id, period_start, period_end, known_at, defaults, denominator, mature, default_definition.

`selected_cohort_ids`

Explicit unique cohort identifiers.

`knowledge_date`

ISO calendar information cutoff.

`weighting`

annual_mean or pooled_obligor.

`uncertainty`

none, year_t or clopper_pearson.

`dependence_assumption`

unspecified, independent_years or independent_homogeneous_obligors.

`representativeness`

List with assessment (representative, not_representative or unknown), rationale and reference_ period.

`target`

List with event, default_definition, horizon=1, time_unit=years, probability_measure=PHYSICAL, conditioning.

`confidence_level`

Confidence level in (0,1).

`context`

Optional calculation context.

Details

Periods are start-inclusive, end-exclusive calendar years; February 29 anniversaries use February 28. Annual mean equally weights cohort rates. Pooled weighting divides total defaults by total denominator. Year-t intervals require independent nonoverlapping years and at least two cohorts;

they are approximate and intersected with [0,1]. Exact binomial intervals require independent homogeneous obligors and nonoverlapping periods. Representativeness and independence remain unverified user assumptions. Repeated obligors and dependent defaults require other uncertainty methods. An observed long-run rate is not automatically TTC.

Value

CalcResult with value list pd, lower, upper and cohort, target, weighting and assumption diagnostics. Bounds are NULL when uncertainty is none.

References

Clopper and Pearson (1934), as documented in `stats::binom.test`.

[1.5in]

`estimate_simulation_risk`
Sampling Aware Default and Joint Loss Estimates

Description

Estimate risk for one initially performing subject, keeping exact IID binomial uncertainty separate from weighted or dependent scenario summaries.

Usage

```
estimate_simulation_risk(path_records, sampling_contract,
  confidence_level = 0.95, context = NULL)
```

Arguments

`path_records`

Records with unique scenario_id, subject_id, logical initial_default, defaulted and horizon_complete, nonnegative weight and ead_at_default, and signed economic_loss. Nondefault rows have zero default EAD and loss. Initially defaulted rows are excluded. Other positive-weight rows must have a complete target horizon.

`sampling_contract`

List with contract_id, version, subject_id, currency, probability_measure (PHYSICAL/RISK_NEUTRAL), positive horizon, time_unit (days/months/years), conditioning (initially_performing), event (first_default), and sampling_design (IID_UNWEIGHTED/FIXED_WEIGHTED_SCENARIOS/DEPENDENT_OR_QMC).

`confidence_level`

Two-sided exact binomial confidence level in (0,1).

`context`

Optional compatible calculation context.

Details

IID estimates require unit weights and return a Clopper–Pearson interval. The zero-default upper bound is positive. A zero plug-in standard error is not certainty. Loss Monte Carlo standard error uses the IID sample variance and does not establish tail robustness. Weighted scenarios and dependent or QMC points receive no invented IID confidence interval. Kish effective weight count is a concentration diagnostic, not a substitute binomial trial count. Expected loss is the mean joint default loss. Conditional mean EAD and conditional mean LGD generally do not factor into it; EAD-weighted LGD provides the monetary factorization when defined. A zero-EAD default leaves individual LGD undefined. Economic losses and LGDs are not capped. Censored positive-weight performing trials are rejected instead of silently dropped.

Value

A CalcResult with PD, its applicable interval, expected loss, conditional loss/EAD/LGD, EAD-weighted LGD, sample standard errors where justified, effective weight count, excluded initial defaults, and explicit empty-risk-set status.

References

Clopper and Pearson (1934), Biometrika 26(4), 404–413, doi:10.1093/biomet/26.4.404. See also https://stat.ethz.ch/R-manual/R-devel/library/stats/html/binom.test.html.

[1.5in]

`estimate_tranche_credit_metrics`
Definition-specific tranche risk and paired structure stress

Description

Separates payment-event probability, ultimate capital loss and discounted contractual cash shortfall; compares explicit structures on shared scenarios.

Usage

```
estimate_tranche_credit_metrics(tranche_paths, target,
    discount_factors,
  confidence_level = .95, context = NULL)
stress_tranche_structure(baseline_paths, alternatives, target,
  discount_factors, confidence_level = .95, context = NULL)
```

Arguments

`tranche_paths`, `baseline_paths`

Bound TrancheWaterfallPaths artifact.

`target`

Explicit event TIMELY_INTEREST, LEGAL_MATURITY_PRINCIPAL or ULTIMATE_PRINCIPAL, full calendar-day horizon, days time unit, physical measure and matching conditioning.

`discount_factors`

Named list of positive origin-based factors for all payment dates.

`confidence_level`

Interior confidence level for IID exact binomial intervals.

`context`

Optional compatible target/currency context.

`alternatives`

Rows with unique alternative_id, evaluated tranche_paths and declared assumption. Scenario axes, weights, origin, currency and shared innovation hash must match the baseline.

Details

Discounted shortfall is the positive part of principal promised at legal maturity plus coupons due on actual opening balances minus dated receipts. Signed components are also returned. This includes payment timing and differs from undiscounted principal loss. Loss outside the selected event remains explicit. Open runoff leaves ultimate loss and discounted shortfall unidentified.

Exact Clopper-Pearson intervals and sample-mean discounted loss standard errors apply only to declared IID equally weighted scenarios. Fixed weighted or dependent/QMC scenarios have no IID uncertainty. Conditional principal severity is tied to the selected event; zero event probability yields null severity.

Stress outputs amount and rate deltas with changed notionals disclosed. Different joint default mappings may share identical innovations. Individual scenario-risk intervals are not paired-difference confidence intervals. No causal or optimization claim is made.

Value

CalcResult containing bound TrancheCreditMetrics or TrancheStructureStress with source and innovation bindings.

References

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/binom.test.html

[1.5in]

`expected_loss`
Expected Default Loss

Description

Computes undiscounted expected default loss as PD times LGD times EAD.

Usage

```
expected_loss(pd, lgd, ead, context = NULL)
```

Arguments

`pd`

Finite probabilities in the closed interval from zero to one.

`lgd`

Finite economic loss ratios. Negative ratios and ratios above one are permitted.

`ead`

Finite nonnegative exposures in a consistent monetary unit.

`context`

Optional calculation context.

Details

All inputs are scalar or one-dimensional numeric vectors. Length-one inputs expand; all other lengths must agree. Empty inputs and implicit vector recycling are rejected. The horizon is the horizon of the supplied PD. LGD and EAD must be deterministic or jointly specified conditional on default. Multiplying separately estimated marginal means is generally invalid when residual dependence remains. No currency is inferred and LGD is not capped. Binary exponent scaling preserves representable products when a naive intermediate multiplication would underflow; non-finite final results are rejected.

Value

A `CalcResult` whose `value` is the expected loss, in the EAD monetary unit.

Examples

```
expected_loss(0.02, 0.4, 100000)$value
expected_loss(c(0, 0.1, 1), 0.5, 200)$value
```

[1.5in]

expert_probability
Documented Expert Probabilities, Agreement and Exact Finite Networks

Description

Fits elicited proper Beta priors, updates compatible observations, pools distributions, applies joint likelihood evidence, diagnoses judgments, and performs bounded exact Bayesian-network variable elimination.

Usage

```
fit_elicited_beta_prior(elicitation, elicitation_contract,
  max_cdf_error = .01, context = NULL)
update_expert_beta_binomial(prior, defaults, exposures,
  observation_contract, predictive_exposures = 0,
  credible_level = .95, context = NULL)
pool_expert_probability_distributions(components, weights,
  support, rule, exclusion_reasons = NULL, context = NULL)
update_odds_from_evidence(prior_pd, evidence_groups,
  dependence_contract, previous_evidence_ids = NULL, context = NULL)
check_expert_preference_consistency(comparisons, items,
  incomplete_policy, tolerance = 1e-8, context = NULL)
assess_expert_rating_agreement(first_ratings, second_ratings,
  ordered_grades, rater_pair, weighting, missing_policy,
  bootstrap_indices = NULL, confidence_level = .95, context = NULL)
compile_discrete_bayesian_network(nodes, network_contract,
  elimination_order = NULL, max_factor_cells = 1000000,
    context = NULL)
infer_discrete_credit_network(compiled_network, evidence,
  query_node = NULL, context = NULL)
```

Arguments

`elicitation`

List with method `moments`, interior mean and exactly variance or concentration; or method `quantiles`, increasing interior quantiles/probabilities and positive weights.

`elicitation_contract`

Population ID, reference, known-at ISO date, distinct expert/evidence IDs and explicit default target.

`max_cdf_error`

Maximum acceptable quantile CDF residual, strictly between zero and one.

`context`

Optional calculation context, consistent with any declared default target.

`prior`

Sealed Beta artifact returned by elicitation or a previous posterior.

`defaults`, `exposures`

Integer compatible counts, zero to one million, defaults no larger than exposures.

`observation_contract`

Matching target/population, new evidence IDs, knowledge cutoff, and explicit true conditional-iid and outcome-complete flags.

`predictive_exposures`

Future integer size, zero to 100000, for Beta-binomial predictive probabilities.

`credible_level`

Interior probability for equal-tail posterior credible intervals.

`components`

Matrix whose rows are normalized masses or constant bin densities on common support.

`weights`

Nonnegative opinion weights with positive total; normalized internally.

`support`

List with kind `discrete` and distinct ordered string values, or kind `histogram` and increasing numeric breaks.

`rule`

`linear` or `logarithmic`.

`exclusion_reasons`

Named reasons for every zero-weight component, using zero-based index strings.

`prior_pd`

Prior probability, including zero or one.

`evidence_groups`

List of unique group IDs, distinct evidence IDs, finite positive joint likelihood ratio and reference.

`dependence_contract`

With multiple groups, explicit conditional independence under both default and survival, and a supporting reference.

`previous_evidence_ids`

Previously applied evidence, forbidden from reuse.

`comparisons`

Positive reciprocal square comparison matrix. Missing pairs must be missing symmetrically. Diagonal entries are one.

`items`

Ordered distinct item labels.

`incomplete_policy`

`error` or `connected_components`; disconnected priorities have no identified common scale.

`tolerance`

Positive log-ratio consistency tolerance, not a universal regulatory threshold.

`first_ratings`, `second_ratings`

Aligned grade vectors or lists; NULL/NA represents missingness.

`ordered_grades`

Complete ordered grade labels, including unused categories.

`rater_pair`

Two distinct rater IDs.

`weighting`

`unweighted`, `linear` or `quadratic` similarity weights.

`missing_policy`

`error` or `pairwise_complete`.

`bootstrap_indices`

Optional matrix of zero-based paired-complete resamples, at least two rows, each retaining sample size.

`confidence_level`

Interior probability for paired bootstrap percentile intervals.

`nodes`

Lists with node ID, states, parents, origin (ELICITED/ESTIMATED/FIXED), reference, known-at date and complete conditional probability table.

`network_contract`

Network ID/version, population, as-of/known-at dates, default node/state and explicit default target.

`elimination_order`

Every node exactly once; default reverse lexical topological order.

`max_factor_cells`

Positive maximum CPT or intermediate factor size, at most ten million.

`compiled_network`

Sealed compiled finite network artifact.

`evidence`

Named observed node states; an empty list means no evidence.

`query_node`

Queried node, defaulting to the declared default node.

Details

Targets contain event `default`, positive horizon, time unit (days/months/years), PHYSICAL or RISK_NEUTRAL measure, and conditioning. Elicitation and observations retain distinct evidence. Moment priors use alpha=mean*concentration and beta=(1-mean)*concentration. Quantile fitting minimizes weighted squared CDF residuals in log shapes bounded by [-18,18], using native nlminb from starts (1,1), (2,18), (18,2), with residual, boundary and local-Jacobian identification checks. Prior concentration is not an empirical count.

Conjugate updates add defaults to alpha and nondefaults to beta. Posterior mean, median, credible interval and prior weight are separate. Future count probabilities integrate binomial probabilities over the posterior Beta. Existing evidence IDs cannot be reused; current prior availability cannot be later than observations. Complete conditionally iid sampling is a supplied assumption, not an inferred property.

Linear pools average normalized densities. Logarithmic pools exponentiate weighted log densities and normalize on declared finite support. Disjoint positive-weight support is an error. Histogram densities are constant in bins; cumulative output is at right endpoints. Opinion correlation is not independent evidence.

Likelihood updates add joint log likelihood ratios. Distinct groups require conditional independence under both outcomes. Zero/one priors remain dogmatic under finite positive likelihood ratios.

Preference priorities solve log least squares under a zero-sum log-priority constraint in each connected component. Pair and triangle residuals and DFS cycle witnesses diagnose rather than repair inconsistency. Uncertainty is judgment disagreement only; no sampling interval is asserted.

Weighted kappa compares observed agreement with independent empirical rater marginals. Expected agreement within 1e-15 of one is degenerate. Optional paired percentile bootstrap assumes independent cases unless caller supplies a valid cluster resampling design. Any degenerate replicate suppresses the interval instead of being silently discarded. Agreement does not establish predictive validity.

Networks require every parent-state combination once, nonnegative CPT rows normalized within 1e-12, an acyclic graph and dated parameter provenance. Exact log-domain factor multiplication and elimination marginalize absent evidence. Querying an observed node still checks evidence likelihood. Impossible evidence errors; positive likelihood underflow is separately flagged with finite log likelihood. Every intermediate obeys the frozen size limit. These are observational probability networks, not automatically causal models.

Value

A CalcResult whose value holds the sealed distribution/network, posterior/count distribution, pool, evidence contributions, preference diagnostics, agreement matrix/kappa, or exact queried probabilities and likelihood. Hashes identify frozen content, not authenticated authorship.

References

SHELF: https://shelf.sites.sheffield.ac.uk/. R Beta distribution: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Beta.html. De Carvalho et al., logarithmic pooling: https://arxiv.org/abs/1502.04206v2. Crawford and Williams (1985), A note on the analysis of subjective judgment matrices, Journal of Mathematical Psychology 29, 387–405. Revelle, weighted kappa: https://personality-project.org/r/psych/help/kappa.html. Scutari, exact inference: https://www.bnlearn.com/about/teaching/eng-apm-inference.pdf.

Examples

```
target <- list(event="default", horizon=1, time_unit="years",
  probability_measure="PHYSICAL", conditioning="performing")
contract <- list(population_id="synthetic", reference="synthetic exercise",
  known_at="2024-01-01", expert_ids=list("one"),
  evidence_ids=list("elicitation-one"), target=target)
prior <- fit_elicited_beta_prior(
  list(method="moments", mean=.1, concentration=20), contract)
prior$value$alpha
```

[1.5in]

`expert_qualitative`
Frozen Qualitative Rubrics and Dated Structured Evidence

Description

Compiles explicit categorical or numeric qualitative criteria and evaluates known, valid structured facts with evidence and conflict traces. No narrative inference or automatic conversion of points to PD occurs.

Usage

```
compile_qualitative_rubric(criteria, rubric_contract, score_policy,
  interactions = NULL, rating_limits = NULL, context = NULL)
evaluate_qualitative_evidence(compiled_rubric, observations,
    evidence,
  as_of, known_at, conflicts = NULL, context = NULL)
```

Arguments

`criteria`

Criterion lists with ID, kind, nonnegative weight, evidence requirements, missing/non-applicable policies and distinct scored levels.

`rubric_contract`

Rubric ID/version, population, reference, known-at date, score direction higher-is-worse, ordered grades and explicit default target.

`score_policy`

Ordered finite score floor/cap, minimum weighted evidence coverage, and unresolved-interaction policy refer or apply-adverse.

`interactions`

Optional IDs, AND conditions mapping criteria to allowed levels, nonnegative adverse point deltas and references.

`rating_limits`

Optional IDs, AND level conditions, best/worst allowed grades and references.

`context`

Optional calculation context consistent with the rubric target.

`compiled_rubric`

Sealed rubric returned by compilation.

`observations`

Dated records with observation/criterion IDs, value, non-applicable Boolean, observation time, knowledge/validity fields and evidence IDs.

`evidence`

Dated evidence with ID/type, source ID, declared independence group, content fingerprint and VERIFIED/UNVERIFIED/REJECTED status.

`as_of`, `known_at`

ISO observation and knowledge cutoff dates.

`conflicts`

Optional dated conflict records linking observations of one criterion, OPEN/RESOLVED status, reference and selected resolution observation when resolved.

Details

Criteria weights are normalized once at compilation and never renormalized because evidence disappears. Categorical levels must exactly partition declared categories. Numeric levels must be finite contiguous disjoint intervals, left closed/right open except the last upper endpoint, with points

respecting any declared increasing or decreasing risk order. Higher points always indicate greater risk.

Each criterion declares required evidence types and a positive minimum number of independent source groups. Independence is a supplied assertion, not inferred from document counts. Missing policy is refer or worst. Non-applicable policy is separately refer, worst or fixed; fixed points must be within the criterion range and require valid supporting evidence. Unknown facts never become good or average observations.

Only verified, known and valid evidence supports facts. Validity intervals are half open. Observation time cannot exceed its knowledge time. Unknown/future/expired records are explicitly excluded. Multiple active identical facts can combine evidence; differing facts always refer until an explicit known valid resolution. Open conflicts, incompatible resolutions and unavailable resolution selections remain unresolved.

Weighted criterion points are summed, nonnegative adverse interaction deltas added and declared floor/cap applied. Unknown contributions produce an interval and no point estimate. Unknown interactions either refer or conservatively apply the adverse delta. Missing/assumed and non-applicable records do not receive invented levels. Active rating limits intersect; inconsistent or unresolved limits refer. Points are not mapped automatically to PD or grade.

Evidence coverage uses original normalized criterion weights. A conservative numerical score may coexist with REFER when evidence coverage or judgment remains unresolved; that score does not authorize a credit action. Content hashes record exact rubric and sorted input-record content, not author authentication.

Value

A CalcResult containing the sealed rubric, or score and interval, readiness/referral status, evidence coverage, criterion and interaction contributions, rating limits, conflicts, record exclusions, input hashes and declared cutoffs/target.

References

SHELF structured elicitation background: https://shelf.sites.sheffield.ac.uk/. These functions implement explicit local contracts and do not claim an elicitation workshop occurred.

[1.5in]

`explainability_attribution`
Frozen Linear and Grouped Shapley Credit Attribution

Description

Exact linear-predictor contributions and exact or permutation grouped Shapley attribution with explicit reference distributions.

Usage

```
explain_linear_credit_prediction(linear_model, transformed_inputs,
    baseline,
  feature_groups, offset = NULL, baseline_offset = 0,
    context = NULL)
estimate_credit_shapley_values(predictor, record, background,
    model_contract,
  feature_groups, masking_contract, method = "exact",
    permutations = NULL,
  max_coalitions = 4096, context = NULL)
```

Arguments

`linear_model`

Frozen model contract plus coefficients, intercept, link (identity, logit, probit, cloglog) and transformed_feature_lineage, a named list of raw-source-name lists.

`transformed_inputs`

Numeric row-by-feature matrix already using frozen transformations.

`baseline`

Numeric reference feature vector.

`feature_groups`

Named list partitioning every transformed feature exactly once into groups. Group IDs are sorted for coalition indexing.

`offset`

Optional numeric row offsets.

`baseline_offset`

Scalar reference offset.

`predictor`

Deterministic local function from numeric row matrix to numeric predictions or CalcResult with numeric value. No fitting takes place.

`record`

Numeric feature vector to explain.

`background`

Interventional empirical reference list with values (matrix or row lists) and weights. NULL for supplied conditional distributions.

`model_contract`

model_id, model_version, ordered feature_order, output_scale (probability, log_odds, score, currency), deterministic_frozen TRUE and target with event, horizon, time_unit, probability_measure and conditioning. Use explicit lists for portable content hashes.

`masking_contract`

masking, reference_id and reference_version. Masking is INTERVENTIONAL_EMPIRICAL or SUPPLIED_CONDITIONAL_DISTRIBUTIONS. Conditional coalition_backgrounds maps EMPTY or comma-separated zero-based sorted-group indices to values/weights distributions. Observed coalition features must equal the record. Permutation mode declares permutation_design FIXED_PERMUTATIONS or IID_RANDOM_PERMUTATIONS.

`method`

exact or permutation.

`permutations`

Explicit list of complete group-ID permutation lists, required for permutation mode.

`max_coalitions`

Positive integer budget for evaluated coalitions; exact mode requires all powerset coalitions.

`context`

Optional calculation context consistent with the target.

Details

Linear contributions are coefficient times displacement from the reference and add on the linear predictor scale. Offsets are separate. Nonlinear response and response differences are shown separately; logit contributions are not additive PD changes.

Interventional masking breaks dependence between observed and masked features. Conditional distributions are supplied by the user and require independent validation. Exact Shapley weights are factorial subset weights. Permutation contributions telescope and are averaged; only declared IID random permutations with at least two draws receive a sampling standard error. Fixed permutation plans have none. This uncertainty excludes model and reference estimation uncertainty. Local accuracy and complete-coalition prediction consistency are checked. Coalitions are cached within the explicit budget.

For the product function x*y at (2,3) against (0,0), exact contributions are (3,3). With perfectly correlated equal-probability (0,0)/(1,1), explaining f=x conditionally at (1,1) yields (.25,.25) about baseline .5. Attributions explain a supplied reference game, not causal effects or credit approval.

Value

CalcResult containing contributions, reference and prediction, reconstruction/local-accuracy diagnostics and content hashes.

[1.5in]

features
Frozen training transformations and auditable financial features

Description

Estimates preprocessing and supervised binning only from declared training rows, applies frozen artifacts, and constructs units-aware ratios and exact-calendar vintage-aware lags.

Usage

```
fit_preprocessor(train_features, feature_schema, target, training,
  model_id, model_version, context = NULL)
transform_features(features, fitted_preprocessor, context = NULL)
build_financial_ratios(financial_statements, ratio_spec,
    context = NULL)
build_distributed_lags(series, observations, lag_spec, frequency,
  gap_policy = "null", context = NULL)
fit_monotone_bins(x_train, y_train, feature_name, direction,
    min_bin_count,
  max_prebins, target, training, model_id, model_version,
    context = NULL)
compute_woe_iv(bin_counts, smoothing, woe_sign, context = NULL)
```

Arguments

`train_features`, `features`

Nonempty list of named row records or data frame.

`feature_schema`

Ordered specifications with name and type numeric/categorical. Numeric: imputation error/mean/median/constant, scaling none/standard/robust, optional fill_value and missing_indicator. Categorical: missing_policy error/separate/mode, unknown_policy error/separate, optional drop_ reference training category.

`target`, `training`

Common ModelArtifact target and training metadata. Training contains data_fingerprint, split_id, known_at and optional data_origin.

`model_id`, `model_version`

Explicit model identity and version.

`context`

Optional calculation context.

`fitted_preprocessor`

FitResult or ModelArtifact from preprocessing or monotone binning.

`financial_statements`

Rows entity_id, as_of, period_start, period_end, currency, values and units. Values/units are named lists. Period end equals as_of.

`ratio_spec`

Rows name, numerator and denominator named field coefficients, denominator_policy positive/nonzero, output_unit dimensionless.

`series`

Records entity_id, feature, observed_at, available_at, known_at, value. Dates ISO with observed <= available <= known.

`observations`

Query records entity_id, as_of, known_at.

`lag_spec`

Rows name, feature, nonnegative integer lag, operation level/difference/relative_change.

`frequency`

days, months or years.

`gap_policy`

null or error for unavailable lag/current value or zero relative-change base.

`x_train`

Numeric observations with missing values allowed.

`y_train`

Complete observed binary default labels.

`feature_name`

Name of the binned feature.

`direction`

increasing or decreasing observed bin risks.

`min_bin_count`, `max_prebins`

Positive integer minimum nonmissing support and maximum initial approximate count bins.

`bin_counts`

Rows with unique bin_id, defaults and nondefaults as nonnegative integer counts.

`smoothing`

Nonnegative additive class-conditional smoothing. Positive smoothing needed for zero class cells; both classes must exist before smoothing.

`woe_sign`

nondefault_over_default or default_over_nondefault.

Details

Standard scaling uses population standard deviation after imputation; robust scaling uses median and type-7 interquartile range. Zero scale becomes one and is reported. Missing and unknown categories have distinct declared handling; categorical mode ties are lexical. One-hot output names and lineage are frozen. Constant columns are retained and require explicit downstream model treatment.

Monotone bins preserve ties, build approximate count prebins, merge small bins with the nearest-risk neighbor (left on ties), then pool adjacent monotonicity violations. This is not claimed globally optimal. Missing has a separate bin exempt from support and monotonicity. Right-closed boundaries use observed maxima; the final NULL upper boundary has no upper limit. Transformation returns bin IDs, not WOE. WOE uses class-conditional smoothed shares; IV is sign-invariant and equals the sum of contributions.

Ratios require aligned statement dates and matching input units, with explicit missing/invalid-denominator statuses instead of infinities. Lags select actual calendar dates and the latest available vintage at the query information cutoff. Month/year lags clamp month ends. Future revisions cannot alter historical-cutoff replay.

Value

Fits return FitResult with ModelArtifact. Transform returns CalcResult containing design_matrix, feature_order and lineage with coverage diagnostics. Ratios and lags return row records with values, statuses and eligibility. WOE returns bin shares, weights of evidence, IV contributions and total IV.

References

Pool-adjacent-violators principle: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.isotonic_regression.html. All support, unit and vintage policies are explicitly declared library contracts.

[1.5in]

`find_feasible_credit_counterfactual`
Finite-Domain Feasible Credit Counterfactual Search

Description

Enumerate an explicit candidate domain under structural constraints and declared costs, preserving search limits and failed candidates.

Usage

```
find_feasible_credit_counterfactual(predictor, record, candidates,
  model_contract, feasibility_contract, cost_contract, target,
    candidate_ids,
  max_candidates = 10000, policy_evaluator = NULL,
    policy_contract = NULL,
  context = NULL)
```

Arguments

`predictor`

Frozen local function producing one scalar prediction per row.

`record`

Numeric baseline vector in model feature order.

`candidates`

Finite candidate matrix or row lists in model feature order.

`model_contract`

Frozen model identity, feature order, output scale and target as for ICE.

`feasibility_contract`

Explicit bounds, immutable/actionable partition, integer, one-hot, linear and derived constraints as for ICE.

`cost_contract`

cost_id, version, metric, feature_weights and feature_scales. Both named maps cover every feature. Weights are nonnegative with positive total, scales positive. Metric is WEIGHTED_L1, WEIGHTED_L2 or WEIGHTED_CHANGE_COUNT.

`target`

target_id, version and kind. MODEL_OUTPUT specifies operator <=, >= or INTERVAL and scalar value or two closed endpoints. POLICY_ACTION specifies an actions list.

`candidate_ids`

Distinct candidate identifiers.

`max_candidates`

Positive integer examination budget, including infeasible candidates.

`policy_evaluator`

Optional full-policy replay function required for policy-action targets; follows the ICE binding contract.

`policy_contract`

Frozen policy hash and fixed-source bindings as for ICE.

`context`

Optional common calculation context.

Details

For changes divided by feature scales, L1 sums weighted absolute changes, L2 takes the square root of weighted squared changes, and change-count sums weights for exactly changed features. Candidates are visited by cost then ID. Infeasible candidates are reported without prediction. Every evaluated failure remains visible. A complete finite search reports FEASIBLE_SOLUTION or NO_FEASIBLE_SOLUTION. SEARCH_LIMIT remains the status whenever candidates remain unevaluated, even if a feasible witness exists. Optimality concerns only the supplied finite domain and declared costs, never continuous or causal global optimality. Policy targets replay full gates rather than equating low risk to approval. No real credit action is executed.

Value

CalcResult with search status, best witness, all evaluated/successful candidates, unevaluated IDs, domain scope, baseline and contract hashes.

See Also

`compute_credit_ice`

[1.5in]

`fit_binary_glm`
Estimate and Apply Binary Probability Models

Description

Fits an unpenalized Bernoulli likelihood natively in R and applies frozen coefficients using an explicit target and feature contract.

Usage

```
fit_binary_glm(X_train, y_train, link, feature_order, fit_intercept,
  target, training, model_id, model_version, weights = NULL,
  weight_type = "frequency", offset = NULL, max_iter = 100L,
  tolerance = 1e-10, context = NULL)
predict_pd(fitted_model, X, feature_order, expected_target,
  offset = NULL, context = NULL)
```

Arguments

`X_train`, `X`

Training uses a finite numeric matrix; prediction accepts numeric matrices or numeric row lists, typed categorical tables for naive Bayes, X/groups lists for grouped Bayesian models, or dated masked records for GRU/LSTM. A zero-column numeric matrix specifies an intercept-only model. Features must match the frozen transformation contract.

`y_train`

Observed numeric zero/one outcomes, one per training row.

`link`

Explicit `"logit"`, `"probit"` or `"cloglog"`.

`feature_order`

Distinct nonempty names in exact matrix-column order; empty for a zero-column matrix.

`fit_intercept`

Explicit logical indicating whether to add an intercept.

`target`, `expected_target`

Named list with event, positive horizon, time unit, probability measure and conditioning. Prediction requires the exact fitted target.

`training`

Named list with data fingerprint, split identifier and ISO knowledge date. Optional data origin declares synthetic or user-supplied observations.

`model_id`, `model_version`

Nonempty model identity strings.

`weights`

Nonnegative integer frequency counts per row; default one. Zero-weight rows are excluded.

`weight_type`

Only `"frequency"` is supported.

`offset`

Optional known additive link-scale vector. Its presence at prediction must match fitting.

`max_iter`

Positive integer IRLS iteration limit.

`tolerance`

Positive convergence tolerance below one.

`context`

Optional calculation context, with compatible horizon, measure and conditioning.

`fitted_model`

Supported frozen ModelArtifact, FitResult, or model-loading CalcResult. Supported families include GLM, discriminant, naive Bayes, Firth, penalized/ordinal, GAM, forests, boosting, Bayesian, MLP, GRU/LSTM, scorecards, stacked models, selection protocols, joint-selection models and market-to-physical bridges; raw SVM margins remain uncalibrated scores.

Details

The estimator uses `stats::glm.fit`. No preprocessing, feature selection or calibration is inferred. Integer frequency weights mean repeated identical observations, not sampling, class-balancing or EAD weights. Covariance is inverse expected Fisher information with independent Bernoulli observations and unit dispersion. It is not cluster-robust. AIC and BIC use the Bernoulli likelihood, with BIC sample size equal to the sum of frequency counts.

The design must have full rank at relative singular-value tolerance 1e-10. A bounded linear program checks complete and quasi separation: maximize mean signed margin with every signed margin nonnegative and each direction coefficient between minus one and one, after column normalization. Mean margin above 1e-8 is rejected; feasibility tolerance is 1e-7. Nonconvergence, unusable information, excessive scaled likelihood score or fitted numerical probability boundaries raise controlled errors. No penalized fallback occurs. Offsets are supplied anew for prediction. Unsupported models or preprocessing are rejected. Generic prediction checks method/parameter-kind agreement, complete target and feature order; a supplied context knowledge date must not precede model availability. Common diagnostics bind canonical model/input hashes, training fingerprint/date, target and prediction count. Typed categorical, sequence and grouped inputs are preserved. No empirical fit or implicit calibration is performed, and PIT outputs are never relabelled TTC.

Value

Fitting returns a `FitResult` with `model` and `value$model`, including coefficients, expected-Fisher covariance, target, training, provenance and diagnostics. Prediction returns a probability-valued `CalcResult`. Fit diagnostics include likelihood, deviance, AIC/BIC, frequency count, rank, condition number, score and its threshold, iteration count and warnings.

References

R Core Team, https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html; lpSolve, https://cran.r-project.org/package=lpSolve.

Examples

```
target <- list(event = "default", horizon = 1, time_unit = "years",
  probability_measure = "PHYSICAL", conditioning = "performing")
training <- list(data_fingerprint = "analytic-counts", split_id = "train",
  known_at = "2025-12-31", data_origin = "SYNTHETIC")
fit <- fit_binary_glm(matrix(numeric(), 2, 0), c(0, 1), "logit",
  character(), TRUE, target, training, "intercept-example", "1",
  weights = c(7, 3))
predict_pd(fit, matrix(numeric(), 1, 0), character(), target)$value
```

[1.5in]

hybrid
Compatible PD Pooling, Held-Out Stacking and Frozen Model Graphs

Description

Combines aligned probability estimates, fits an actual held-out penalized logistic meta-model, executes a typed frozen graph and explains its result relative to explicit baseline inputs.

Usage

```
combine_pd_signals(pd_signals, weights, combination_scale,
  combination_contract, context = NULL)
fit_stacked_pd(out_of_fold_base_predictions, outcomes,
    base_model_ids,
  split_plan, meta_model_spec, target, training, model_id,
    model_version,
  weights = NULL, context = NULL)
evaluate_model_graph(model_graph, input_data, parameter_snapshots,
    context)
decompose_hybrid_prediction(hybrid_result, baseline_inputs,
  decomposition_scale = "native", context = NULL)
```

Arguments

`pd_signals`

Records with signal/model IDs, model version, ordered row IDs, PD vector, target, observation and knowledge dates. All signals must align exactly.

`weights`

Nonnegative global pool weights, row-specific regime weights for mixtures, or optional aligned meta-training frequency weights for stacking.

`combination_scale`

Linear, log_odds or mixture.

`combination_contract`

Interpretation estimator_pool, log_odds_pool or latent_regime_mixture, plus reference. Mixture requires ordered regime IDs and conditional_regime_probabilities TRUE.

`context`

Calculation context. Graph execution requires explicit dated context; decomposition preserves the original context.

`out_of_fold_base_predictions`

Row-by-base probability matrix; missing predictions allowed only outside used held-out rows.

`outcomes`

Aligned outcomes. Used training and held-out labels must be mature, known and binary. Unused outer test labels are not inspected.

`base_model_ids`

Ordered distinct base IDs.

`split_plan`

Calendar-clamp horizon rule, rows with IDs/entity/group/role and observation, label, coverage and prediction dates; forward folds with training/validation IDs, fit date and base-specific provenance.

`meta_model_spec`

Input scale probability or log_odds and actual fit_penalized_glm arguments: penalty, lambda path, selected lambda, intercept and scaling, optional elastic-net ratio and solver controls.

`target`

Default event, positive horizon, days/months/years unit, physical or risk-neutral measure and conditioning.

`training`

Training data fingerprint, split ID and knowledge date; optional data origin.

`model_id`, `model_version`

Explicit fitted artifact identity.

`model_graph`

Graph ID/version, nodes and output node. Each node has ID, operation, named input ports, parameters and output contract.

`input_data`

Named input snapshots with value, contract and optional already-applied effect IDs.

`parameter_snapshots`

Named frozen model/calibrator artifacts.

`hybrid_result`

Sealed evaluated graph, including input/parameter snapshots and trace.

`baseline_inputs`

Exactly matching input snapshots with only values changed.

`decomposition_scale`

Native output scale or log_odds for interior PD outputs.

Details

Pools normalize nonnegative weights. Log-odds pooling preserves finite-weight zero/one limits, rejects simultaneous positive-weight zero and one, and never clips. Mixtures use supplied row-specific conditional regime probabilities; they are not event unions. Mixture within- and between-regime Bernoulli variance sum to p(1-p). Dispersion is not uncertainty.

Stacking validates forward-time, entity and group separation. Every base lineage declares model ID/version, target, held-out origin, reference, training fingerprint, known-at and fit/preprocessing/selection row IDs. These must lie inside fold training rows. Base fits precede validation forecasts; known prediction dates cannot exceed forecasts. Used labels must be known and full horizons mature even for early known defaults. Negative labels require declared complete-through coverage. Date horizons use integer days or whole months, clamping month-end dates. Outer development/calibration/test entities and groups are disjoint. Only held-out development rows train the meta-model. Declared lineage is checked, not authenticated. Penalties are caller selected; no test-based selection occurs.

The meta-model uses actual native penalized binomial likelihood. Positive-weight constant bases are removed and frozen; all-constant models fit a real intercept and prediction flags deviations from the ignored constant training values. Log-odds inputs must be interior. No post-selection uncertainty is claimed. Use predict_pd with base-model IDs as feature order on the returned fitted artifact.

Graph contracts contain unit, target, ordered row IDs, as_of and known_at. Units are design, probability, loss_fraction and money. Designs add feature_order; money adds currency. Vectors exactly match rows without recycling. Supported operations are input (input_id), predict_pd (design port, snapshot_id), apply_calibrator (pd port, snapshot_id, base_model_version, optional groups), combine_pd_signals (ordered named signal ports and pooling parameters), log_odds_shift (pd port, aligned finite shifts, effect ID/reference), and expected_loss (pd/lgd/ead ports and loss assumption/reference). Frozen parameters must be known by the cutoff. New fitting, external state access, unknown operations/parameters and graph cycles are rejected. Financial feedback networks require their domain solver before insertion as typed inputs.

Expected loss requires performing PD, default-conditional LGD/EAD and matching event/horizon/measure/rows/dates, nonnegative monetary EAD and explicit deterministic_ conditional_amounts or conditional_independence_EAD_LGD assumption. Raw economic LGD may exceed one or be negative. This product is not the expectation of generally dependent random exposure and severity. Repeated declared effects in ancestry or across loss legs are rejected. Identical computation content reuses a local cache. All snapshots, node results, effects and cache keys are retained with a content seal.

Decomposition verifies the seal and replays nodes with relative tolerance 1e-12 plus absolute 1e-14 for native numerical differences, with exact contracts and trace relationships. Each contribution changes one whole input from the common baseline, holding others at baseline. Final equals baseline plus those changes plus an explicit interaction remainder. These are neither causal effects nor Shapley values. Local native deltas for same-contract single-parent steps expose calibration/adjustment effects; they must not be summed across branches. Log-odds decomposition requires interior final probabilities in every scenario. Content hashes do not authenticate source evidence.

Value

A CalcResult with pooled values and diagnostics; a FitResult for stacking; a sealed graph CalcResult with replayable lineage; or baseline, input contributions, interaction remainder, local steps and exact reconciliation diagnostics.

References

Held-out stacked-generalization contract: https://sklearn.org/1.7/modules/generated/sklearn.ensemble.StackingClassifier.html. Pooling and decomposition implement the explicitly stated algebraic definitions.

[1.5in]

`importance_sample_defaults`
Gaussian Importance Sampling for First Default and Loss

Description

Evaluates a local deterministic default evaluator under a complete Gaussian mean-shift proposal with ordinary density-ratio estimates.

Usage

```
importance_sample_defaults(standard_normal_innovations,
    proposal_shift,
  evaluator, importance_contract, context = NULL)
```

Arguments

`standard_normal_innovations`

Finite trial by independent standard-normal coordinate matrix.

`proposal_shift`

Finite shift vector covering every coordinate.

`evaluator`

Local function returning aligned defaulted and economic_loss vectors, or a CalcResult containing them. Nondefaults have zero default loss. All randomness must enter via the innovation matrix.

`importance_contract`

Versioned contract identifying evaluator, subject, currency, measure, horizon, time unit, initially-performing first-default conditioning, INDEPENDENT_STANDARD_NORMAL base law, MEAN_SHIFT proposal, IID or DEPENDENT_OR_QMC design, and both deterministic/all-randomness declarations.

`context`

Optional matching calculation context.

Details

For x=z+mu the likelihood ratio is exp(-mu'z-mu'mu/2). Freeze the proposal independently of evaluated IID draws. The evaluator defines default, not the sampling transformation. Ordinary finite-sample PD estimates can exceed one and are not clipped. Self-normalized estimates are generally biased. Zero proposal defaults provide no exact target-PD upper bound. QMC/dependent designs have no IID standard error. Finite second moments are required but not established for arbitrary caller loss functions. Density-ratio numerical underflow and overflow are rejected.

Value

CalcResult containing ordinary PD and loss estimates, likelihood ratios and contributions, separate self-normalized PD, weight concentration and applicable IID sample standard errors.

References

Art B. Owen, Monte Carlo theory, methods and examples, Chapter 9. https://artowen.su.domains/mc/Ch-var-is.pdf

---

[1.5in]

`lgd`
Dated Recovery Valuation and Raw Economic Workout LGD

Description

Value known realized recovery/cost cashflows using explicit discount and FX contracts, preserving open-workout incompleteness and raw economic LGD.

Usage

```
discount_recoveries(cashflows, discount_curve, fx_contract,
    default_date,
  evaluation_cutoff, reporting_currency, context = NULL)
workout_lgd(ead_at_default, discounted_net_recoveries,
    episode_status,
  default_date, evaluation_cutoff, currency, context = NULL)
```

Arguments

`cashflows`

Data frame or list of rows with unique cashflow_id, cashflow_date, known_at, currency, kind and nonnegative amount. Kind is RECOVERY, DIRECT_COST or INDIRECT_COST. Dates are ISO calendar dates.

`discount_curve`

Named list with anchor_date equal to default_date, known_at, discount_basis description and points containing cashflow_date/discount_factor. Factors are finite nonnegative exact-date values; the anchor factor, if supplied, is one. No interpolation is performed.

`fx_contract`

Named list with reporting_currency, known_at, conversion_basis (cashflow_date or default_date), and rates containing currency/date/rate. Rates are reporting-currency units per cashflow-currency unit, strictly positive, with explicit rate one for reporting currency.

`default_date`, `evaluation_cutoff`

Default and inclusive evaluation dates. Default cannot follow evaluation. Cashflows occur at or after default, become known no earlier than realization, and are included only if realized and known by evaluation. Curve and FX contracts must be known by evaluation.

`reporting_currency`, `currency`

Explicit nonempty currency, consistent with any calculation context currency.

`context`

Optional calculation context or context result.

`ead_at_default`

Finite nonnegative exposure denominator.

`discounted_net_recoveries`

Finite signed scalar or discount_recoveries CalcResult with matching currency, default date and evaluation cutoff.

`episode_status`

OPEN or CLOSED, as established by caller evidence.

Details

Each contribution is amount times FX times discount factor; costs reduce net recovery. Exact date/currency keys are required, and duplicates are rejected. Future or not-yet-known cashflows are excluded visibly, not replaced with forecasts. The supplied discount and FX contracts define the economic valuation convention; no regulatory rate is supplied.

Loss is EAD minus net recovery. Raw economic LGD can be below zero or above one and is never clipped. An open workout does not establish a final loss merely because data ended. These methods do not estimate remaining recovery. Products use scaling to avoid intermediate overflow; nonfinite final products, sums or ratios fail.

Value

A CalcResult. discount_recoveries returns discounted gross recovery, costs, net recovery, signed individual contributions and excluded cashflows with reasons. Diagnostics retain valuation dates, discount/FX convention and input fingerprint. workout_lgd returns economic_loss, economic_lgd_to_date, economic_lgd, completeness, denominator status and the input exposure/net recovery. Final economic_lgd is NULL for OPEN workouts; both ratios are NULL at zero EAD while monetary loss remains.

Examples

```
workout_lgd(100, 80, "CLOSED", "2024-01-01", "2024-06-30", "EUR")$value
workout_lgd(100, -20, "OPEN", "2024-01-01", "2024-06-30", "EUR")$value
```

[1.5in]

lgd_downturn
Declared Adverse-Scenario Conditional LGD Comparison

Description

Compare matched portfolio covariates under a frozen conditional LGD model and dated scenario policy, optionally with supplied paired model draws.

Usage

```
estimate_downturn_lgd(fitted_lgd_model, baseline_covariates,
    adverse_covariates,
  aggregation_weights, feature_order, expected_target,
    downturn_definition,
  baseline_precision_design = NULL, adverse_precision_design = NULL,
  precision_feature_order = NULL, model_draws = NULL,
  uncertainty_contract = NULL, interval_level = 0.95,
    context = NULL)
```

Arguments

`fitted_lgd_model`

Frozen supported conditional LGD model or FitResult.

`baseline_covariates`, `adverse_covariates`

Matched numeric predictor matrices.

`aggregation_weights`

Nonnegative finite portfolio weights, positive total; normalized by scaled summation. They are not fitting frequency weights.

`feature_order`, `expected_target`

Exact frozen predictor order and target.

`downturn_definition`

Named object with definition_id, version, selection_basis, baseline_scenario_id, adverse_scenario_id, known_at, as_of, ordered unique unit_ids, extrapolation_policy. Policy and model must be known at as_of. Extrapolation policy is error or report for positive-weight marginal training-support violations.

`baseline_precision_design`, `adverse_precision_design`, `precision_feature_order`

Separate optional precision predictors matching the frozen beta model.

`model_draws`

Optional list of at least two compatible frozen model artifacts.

`uncertainty_contract`

For supplied draws: source is bootstrap, posterior or user_supplied; interpretation is a nonempty statement of inferential meaning.

`interval_level`

Central empirical interval coverage in (0,1).

`context`

Optional compatible calculation context; as_of must match policy.

Details

No scenario selection, regulatory surcharge or positivity floor is inferred. A model can imply a negative adverse-minus-baseline difference. Identical scenarios produce zero difference. Marginal training ranges do not establish joint support or causality. Uncertainty is NOT_ESTIMATED unless paired model draws are supplied. Each draw evaluates both scenarios; empirical linear quantiles (type7) summarize baseline, adverse and paired difference, retaining the declared interpretation. No statistical coverage guarantee is created from user-supplied illustrative draws.

Value

CalcResult with baseline_lgd, downturn_lgd, signed difference, unit-level predictions, normalized weights and uncertainty. Diagnostics retain policy/model hashes, explicit policy and baseline/adverse marginal range flags.

[1.5in]

`lgd_models`
Frozen Conditional LGD Mean and Distribution Models

Description

Estimate fractional logit, variable-precision beta and structural zero-and-one inflated beta models, preserving explicit conditional targets.

Usage

```
fit_fractional_lgd(X_train, lgd_targets, feature_order, link,
    covariance,
  target, training, model_id, model_version, weights = NULL,
  weight_type = "frequency", fit_intercept = TRUE, max_iter = 200,
```

```
  tolerance = 1e-10, context = NULL)
fit_beta_lgd(X_train, lgd_targets, feature_order, mean_link,
    precision_link,
  target, training, model_id, model_version,
    precision_design = NULL,
  precision_feature_order = NULL, weights = NULL,
    weight_type = "frequency",
  fit_intercept = TRUE, max_iter = 500, tolerance = 1e-10,
    context = NULL)
fit_zero_one_inflated_lgd(X_train, lgd_targets, feature_order,
    component_spec,
  target, training, model_id, model_version,
    precision_design = NULL,
  precision_feature_order = NULL, weights = NULL,
    weight_type = "frequency",
  fit_intercept = TRUE, max_iter = 500, tolerance = 1e-10,
    context = NULL)
predict_lgd(fitted_model, X, feature_order, expected_target,
    default_condition,
  precision_design = NULL, precision_feature_order = NULL,
  quantile_levels = NULL, context = NULL)
```

Arguments

`X_train`, `X`

Finite numeric matrices with explicitly ordered columns. An intercept-only design is a matrix with zero columns and the required row count.

`lgd_targets`

Closed-workout LGD values. Fractional and inflated models accept [0,1]; beta requires (0,1). Raw values outside support are rejected, never clipped.

`feature_order`

Distinct predictor names, or character() for no predictors.

`link`, `mean_link`, `precision_link`

Explicit logit mean and log precision links.

`covariance`

HC0 or HC1 independent-observation robust covariance.

`target`, `expected_target`

Physical target with event, positive horizon, time_unit and conditioning equal to given_default. Prediction requires exact match.

`training`

Named metadata including data_fingerprint, split_id, known_at and workout_population equal to

closed_defaulted_episodes. Optional data_origin. The caller establishes row-level feature chronology and complete workout outcomes.

`model_id`, `model_version`

Nonempty frozen model identifiers.

`weights`, `weight_type`

Nonnegative integer frequencies, positive total, at most 2^53-1. These represent independent replicated observations.

`fit_intercept`

Whether the mean design contains an intercept.

`max_iter`, `tolerance`

Positive iteration limit and score tolerance in (0,1).

`precision_design`, `precision_feature_order`

Optional separate precision matrix and column names. Precision always has an intercept. NULL means constant precision.

`component_spec`

Named list with boundary_origin="structural" and gate_link="logit"; does not support rounded/censored endpoints.

`fitted_model`

Frozen FitResult or validated declarative model artifact.

`default_condition`

The explicit string given_default.

`quantile_levels`

Optional strictly increasing probabilities inside (0,1). Fractional models reject quantiles because their distribution is unidentified.

`context`

Optional calculation context.

Details

Fractional logit maximizes the Bernoulli quasi likelihood and uses the sandwich covariance. Frequency weights enter the meat linearly; HC1 applies N/(N-k). Endpoints remain unchanged, and constant interior targets can yield zero empirical covariance. Full-rank and finite-MLE separation checks are enforced.

Beta uses shape parameters mu*phi and (1-mu)*phi, with logit mean and log precision. Analytic score and observed information include mean/precision cross derivatives. Covariance assumes independent, correctly specified beta observations. The effective normalized score threshold is

max(sqrt(tolerance),1e-7), recorded in diagnostics. Native R reports gradient evaluations rather than iteration counts. Perfectly explained interior responses lack a finite precision estimate and are rejected.

The inflated model estimates boundary probability b, conditional-one probability o and an interior beta model. Endpoint probabilities are b*(1-o) and b*o; the mean is b*o+(1-b)*mu. All components require positive-frequency support and identifiable fits. Quantiles include the endpoint atoms. Component marginal support violations are reported; these do not establish joint support.

Prediction retains fitted provenance and rejects conflicting context target or model-version fields; currency is checked when declared in the target. Models are native declarative artifacts replayable without fitting code callbacks. Native optimizers can take different iteration counts. No downturn margin, regulatory approval or open-workout censoring correction is implied.

Value

FitResult with coefficients, covariance, support ranges, input fingerprints and fit diagnostics. predict_lgd returns CalcResult with mean_lgd, distribution, optional precision, endpoint probabilities and quantiles, plus model fingerprint and marginal training-range violations.

References

Papke and Wooldridge (1996), Journal of Applied Econometrics 11, 619–632. Cribari-Neto and Zeileis (2010), Journal of Statistical Software 34(2), doi:10.18637/jss.v034.i02. Ospina and Ferrari, Inflated Beta Distributions, https://arxiv.org/abs/0705.0700.

[1.5in]

`lgd_workout`
Residual Terminal Payouts and Open Workout Valuation

Description

Fit a cure and recovery hurdle with censored resolution timing, then value remaining terminal payouts without duplicating known cashflows.

Usage

```
fit_recovery_hurdle(episodes, cashflows, X_train, feature_order,
  stage_spec, target, training, model_id, model_version, currency,
  fit_intercept = TRUE, context = NULL)

estimate_open_workout_lgd(episodes, cashflows,
    fitted_recovery_model,
  X, valuation_contracts, future_cost_cashflows, feature_order,
  expected_target, stage_spec, as_of, currency, extrapolation,
  context = NULL)
```

Arguments

`episodes`

List of episode records with unique episode ID; default date and knowledge date; reference date and knowledge date; feature observation and knowledge dates; reporting currency; default EAD; residual exposure at reference; nullable resolution date and knowledge date. Training also requires observation end. Observed resolutions require numeric zero/one cured and nonnegative cure value.

`cashflows`

Realized cashflow records with unique cashflow ID, episode ID, cashflow date, knowledge date, reporting currency, nonnegative amount, and kind RECOVERY, DIRECT_COST or INDIRECT_COST.

`X_train`, `X`

Numeric covariate matrices aligned to episodes. Empty feature designs have the appropriate number of rows and zero columns.

`feature_order`

Distinct covariate names in frozen order.

`stage_spec`

Exact frozen residual stage contract. See Details.

`target`, `expected_target`

Physical given-default target with event residual_terminal_loss_fraction, currency, time unit days, positive horizon, and horizon_role TIMING_REPORTING_REFERENCE. Expected target must match the fitted target.

`training`

Frozen training metadata with data fingerprint, split ID, known_at and workout_population open_and_closed_defaulted_landmarks.

`model_id`, `model_version`

Nonempty model identifiers.

`currency`

Explicit reporting currency; episode and modeled payout amounts must already use this currency.

`fit_intercept`

Whether cure, positive-payout and positive-amount designs include an intercept. The AFT timing design includes its usual intercept.

`context`

Optional calculation context compatible with target, currency, model version and evaluation date.

`fitted_recovery_model`

Frozen recovery hurdle FitResult or model artifact.

`valuation_contracts`

One per episode: discount_curve, fx_contract, future_resolution_dates, resolution_time_convention RIGHT_ENDPOINT, nullable tail_discount_factor and tail_assumption_reference, and unmodeled_future_costs_assumption NONE_BEYOND_SUPPLIED.

`future_cost_cashflows`

Explicit dated future cost records in reporting currency, known by as_of, with IDs distinct from realized flows. These are unconditional costs, not survival-weighted estimates.

`as_of`

ISO evaluation date.

`extrapolation`

Explicit error, flat or parametric timing extrapolation policy.

Details

The stage requires payout_definition RESIDUAL_TERMINAL_GROSS_RECOVERY_OR_CURE_VALUE; cashflow_pattern SINGLE_TERMINAL_PAYOUT_AFTER_REFERENCE; timing_outcome_independence CONDITIONAL_ON_REFERENCE_COVARIATES; cashflow_completeness COMPLETE_THROUGH_OBSERVATION_END; payout_horizon ULTIMATE; positive_distribution lognormal; timing_distribution weibull, lognormal or loglogistic; a nonnegative cure_residual_value_ratio; and a nonempty landmark_definition. The complete stage object must match prediction. Completeness is a user assertion about supplied histories, not an imputation of unreported payments.

The core target horizon is a resolution reporting reference. It does not truncate ultimate terminal payout. `predict_lgd` returns payout_horizon ULTIMATE and an undiscounted residual loss ratio, not whole-workout discounted LGD or loss at the finite reporting horizon. If cure probability is p, positive recovery probability among noncured cases is q, fixed cure ratio is c and positive log-ratio parameters are mu and variance v, expected terminal value ratio is p*c + (1-p)*q*exp(mu+v/2). Loss is one minus this ratio. Its quantiles respect both the zero-recovery and cure-value atoms and can be negative.

Default and all reference features must be known at the reference landmark. Only resolutions observed by observation end and known at training cutoff count as events. Other episodes are right-censored. Training ratios require positive residual exposure. At least one observed cure, one noncured zero payout and two positive noncured payouts are required, with identified nonseparated designs and positive lognormal dispersion. Observations have unit frequency.

Positive post-reference cash recoveries must occur at terminal resolution. Cure asset value must equal residual exposure times the declared cure ratio and cannot duplicate a terminal cash recovery. Costs are not gross payout targets. The closed-case outcome factorization assumes conditional independence of terminal outcome/amount and resolution timing, and independent right censoring given reference covariates. Model availability and feature definitions must match the prediction stage; later features are not silently substituted.

Open-workout valuation includes known past net flows once. Conditional remaining survival is S(age+t)/S(age). Resolution mass in each grid interval receives its right-endpoint default-anchored discount factor. Supplied future costs are deducted once without inferred costs beyond the schedule. Unknown tail discount value leaves final loss and final LGD null; within-grid amounts remain

available. A supplied conditional average tail discount factor requires an economic assumption reference. Closed cases use actual cashflows and cure value without forecast. Zero EAD leaves the ratio null and preserves monetary loss.

Value

`fit_recovery_hurdle` returns a FitResult with frozen gate models, positive amount regression, censored AFT model, target, stage, provenance and identification diagnostics. `estimate_open_workout_lgd` returns a CalcResult with per-episode known cashflows, cure value, future terminal value within the grid, supplied future costs, unresolved probability and value, economic losses and nullable final LGD.

References

NIST/SEMATECH e-Handbook, Lognormal Distribution: https://www.itl.nist.gov/div898/handbook/eda/section3/eda3669.htm.

lifelines WeibullAFTFitter parameterization: https://lifelines.readthedocs.io/en/latest/fitters/regression/WeibullAFTFitter.html. The specific hurdle factorization and terminal stage are explicit modeling assumptions.

See Also

`predict_lgd`, `discount_recoveries`, `fit_aft_model`, `predict_event_curves`

[1.5in]

`loss_paths`
Joint Path, Exclusive Event and Portfolio Loss Expectations

Description

Integrate losses on explicitly aligned probability scenarios. Preserve joint default, LGD, exposure, discount and currency dependence, with finite partition expectations and discrete portfolio tail allocations.

Usage

```
expected_loss_from_paths(default_indicator, lgd_paths, ead_paths,
  scenario_weights, discount_factors, scenario_ids, time_grid,
  currency, probability_measure, uncertainty = "none",
    context = NULL)
lifetime_expected_loss(marginal_pd, conditional_loss,
    discount_factors,
  time_grid, currency, probability_measure, context = NULL)
event_conditioned_loss(states, event_names, currency,
  probability_measure, context = NULL)
counterparty_expected_loss(default_indicator, net_exposure_paths,
  recovery_paths, scenario_weights, discount_factors,
```

```
    counterparty_id,
  netting_set_id, scenario_ids, time_grid, currency,
    probability_measure,
  uncertainty = "none", context = NULL)
aggregate_portfolio_loss(facility_loss_paths, portfolio_mapping,
  scenario_weights, fx_rates, scenario_ids, reporting_currency,
  probability_measure, quantile_levels, context = NULL)
```

Arguments

`default_indicator`

Finite numeric matrix: scenarios in rows and periods in columns. Entries are zero or one, with at most one first default per scenario.

`lgd_paths`

Finite numeric matrix matching the default matrix. Raw economic LGD may be negative or above one; values are not clipped.

`ead_paths`

Nonnegative finite EAD matrix, matching default indicators.

`scenario_weights`

Nonnegative finite vector of scenario probabilities, one per row. Sum must be within 1e-12 of one; this small residual is reported and weights are normalized. Zero-weight scenarios remain identified.

`discount_factors`

Nonnegative finite scalar, period vector or matching scenario-by-period matrix for joint path methods. For lifetime loss an explicit period vector is required. Factors above one are allowed.

`scenario_ids`

Unique nonempty character vector in global scenario order.

`time_grid`

Positive, strictly increasing period-end times in years.

`currency`, `reporting_currency`

Explicit nonempty reporting currency. Any currency in context must match. No currency conversion occurs except through explicit fx_rates in portfolio aggregation.

`probability_measure`

PHYSICAL or RISK_NEUTRAL, consistent with context. These produce different interpretations; the method does not transform measures.

`uncertainty`

none, or iid for a Monte Carlo standard error using at least two equal-probability paths. Independence is a caller assumption, not established from the supplied values.

`context`

Optional named calculation context or calculation-context result.

`marginal_pd`

Vector of first-default period probability masses in [0,1] whose sum is at most one within 1e-12. These are not cumulative PDs.

`conditional_loss`

Finite vector of joint conditional monetary expectations E[LGD * EAD given first default in period]. Negative economic losses are allowed.

`states`

List of all exclusive Boolean event states, including zero-mass states. Each row contains unique state_id, named logical events, scalar probability in [0,1] and finite scalar conditional_loss. Probabilities sum to one within 1e-12. State masses are used as given, without normalization.

`event_names`

Unique nonempty character vector, at most twelve names. Exactly all 2 to the power length(event_names) Boolean combinations are required.

`net_exposure_paths`

Finite signed exposure matrix for one externally established netting set. Negative values are floored at zero for counterparty loss.

`recovery_paths`

Matching finite recovery matrix in [0,1]. Counterparty LGD is one minus recovery.

`counterparty_id`, `netting_set_id`

Explicit nonempty scalar identifiers.

`facility_loss_paths`

List of facility rows with facility_id, currency, scenario_ids and numeric losses. Each facility must cover exactly the global scenario IDs; local order may differ and is explicitly aligned. Losses are already monetary loss amounts under a common horizon and measure.

`portfolio_mapping`

List of facility_id, portfolio_id, share rows. Shares are finite scalar fractions in [0,1]. Repeated facility/portfolio pairs are rejected; total allocation of any facility cannot exceed one within 1e-12. Unallocated shares are returned.

`fx_rates`

Named list by each facility currency. A positive scalar or scenario vector supplies reporting-currency units per local-currency unit. Scenario vectors use global scenario order. Reporting-currency FX must equal one.

`quantile_levels`

Nonempty vector of distinct levels strictly between zero and one. Level order is retained.

Details

For scenario s and period t, joint path loss is the sum of D(s,t) * LGD(s,t) * EAD(s,t) * DF(s,t). The probability-weighted mean integrates all supplied dependencies before aggregation. At most one first default prevents counting the same default again in later periods. The conditional independence comparator multiplies conditional mean LGD and conditional mean discounted EAD within each default period, weighted by its default mass. The difference from joint loss is the aggregate within-default covariance contribution; it is not necessarily positive. The counterparty comparator instead separates default from the complete recovery/exposure/discount product under the supplied scenarios.

The lifetime function multiplies marginal first-default mass by conditional joint loss and fixed period discount. Stochastic loss/discount dependence needs joint path integration. The event function integrates a full mutually exclusive partition exactly once, which allows default and transfer or support events to co-occur without counting their overlap twice. Event meanings and state losses remain explicit caller inputs.

Portfolio paths are aligned, converted with scenario FX, allocated, and summed before tail calculation. Weighted VaR uses the left inverse empirical CDF. Expected Shortfall averages exactly the upper (1 - level) probability mass: all mass strictly above VaR plus only the necessary fraction at VaR. Boundary mass is allocated proportionally across tied scenarios, so facility tail contributions sum to portfolio ES. Zero-weight scenarios do not affect the CDF. The result is not the sum of standalone facility VaRs or a simple conditional mean given loss greater than or equal to VaR.

Products use power-of-two scaling to avoid intermediate overflow when the final product is representable; nonfinite final products or sums fail with controlled NUMERICAL_FAILURE. Native floating-point underflow is still possible. No stochastic paths are generated and no dependence model is inferred by these methods.

Value

A CalcResult with monetary unit and explicit assumptions. expected_loss_from_paths returns expected_loss, path_losses, period_contributions, marginal_default_probability, default_weighted_lgd, default_weighted_discounted_ead, conditional_independence_loss, conditional_lgd_exposure_covariance_contribution and standard_error. Undefined conditional means in zero-default-mass periods are NULL. Diagnostics retain time grid, IDs, measure, residual and canonical input-path fingerprint.

lifetime_expected_loss returns lifetime_expected_loss, period_contributions and total_default_probability. event_conditioned_loss returns expected_loss, state contributions and probability_sum. counterparty_expected_loss extends joint-path results with independent_default_exposure_loss and default_exposure_dependence_contribution; diagnostics retain the netting-set identity and number of negative net values floored.

aggregate_portfolio_loss returns portfolios and unallocated_shares. Each sorted portfolio contains aligned path_losses, expected_loss, named facility_expected_loss and tail_metrics. Each tail includes level, value_at_risk, expected_shortfall and named facility_es_contributions.

Interpretation and limitations

The reported iid standard error measures Monte Carlo integration uncertainty only. It excludes parameter, model and time-discretization uncertainty and is not supplied for unequal scenario probabilities. Finite scenarios can instead represent an explicitly supplied discrete probability law.

All inputs must share their declared scenario meanings, target event, horizon, valuation date and measure. IDs only verify alignment, not economic compatibility. A risk-neutral discounted credit loss is not automatically a complete CVA or capital calculation. Legal netting, collateral close-out, own-default effects and regulatory acceptance are not inferred. Signed raw economic LGD is retained in general loss methods; the counterparty method explicitly uses bounded recovery.

References

Acerbi and Tasche (2002), On the coherence of Expected Shortfall, https://arxiv.org/abs/cond-mat/0104295, Definition 2.6 and equation (3.4), with sign and confidence-level convention translated to upper-tail monetary losses.

Examples

```
r <- expected_loss_from_paths(matrix(c(1, 0), 2, 1),
  matrix(c(0.6, 0), 2, 1), matrix(c(100, 100), 2, 1),
  c(0.5, 0.5), 1, c("default", "survive"), 1,
  currency = "EUR", probability_measure = "PHYSICAL")
r$value$expected_loss
lifetime_expected_loss(c(0.1, 0.05), c(40, 50), c(1, 0.95), c(1, 2),
  currency = "EUR", probability_measure = "PHYSICAL")$value
```

[1.5in]

macro

Release-Aware Macro Dynamics, Cycle Laws and Satellite Transmission

Description

Actual native VAR, Johansen cointegration, Gaussian hidden-regime and dynamic-factor estimation, explicit shared scenario innovations, conditional Gaussian and reverse stress, and dated satellite regressions. Forecast averages do not silently become through-the-cycle estimates.

Usage

```
transform_macro_history(history, variable_transforms, frequency,
    fit_window,
    information_cutoff, context = NULL)
fit_var(transformed_history, lags, deterministic = "constant",
    covariance_estimator = "df_adjusted", training, target,
    model_id,
    model_version, context = NULL)
fit_vecm(level_history, rank, lags,
    deterministic = "constant_inside",
    integration_order = "I1", training, target, model_id,
    model_version,
    context = NULL)
fit_regime_switching(transformed_history, regime_count, lags = 0,
    initialization,
```

```
    max_iterations = 200, tolerance = 1e-08, training, target,
    model_id, model_version, context = NULL)
fit_dynamic_factor(transformed_history, factor_count,
    initialization,
    ar_limit = 0.995, variance_bounds = c(1e-08, 10000),
    max_iterations = 300,
    tolerance = 1e-08, training, target, model_id, model_version,
    context = NULL)
block_bootstrap_scenarios(residual_history, shared_draws,
    factor_ids, scenario_ids,
    time_grid, frequency, block_length, scheme = "moving",
    weights = NULL,
    context = NULL)
simulate_macro_scenarios(fitted_model, innovations,
    initial_state = NULL, regime_uniforms = NULL,
    output_space = "model", sampling_contract = NULL,
    context = NULL)
condition_macro_paths(fitted_model, innovations,
    conditioning_points, initial_state = NULL,
    sampler = "gaussian_equality", sampling_contract = NULL,
    context = NULL)
fit_satellite_model(target_series, macro_history, feature_spec,
    estimator,
    link, split_plan, training, target, target_unit, model_id,
    model_version, weights = NULL, covariance = "classical",
    hac_lags = 0, residual_process = "white_noise", context = NULL)
apply_satellite_model(fitted_satellite_model, scenario_cube,
    idiosyncratic_innovations,
    initial_target_state, context = NULL)
reverse_stress_scenario(fitted_model, loss_model, horizon,
    target_loss, shock_metric,
    initial_state = NULL, shock_bounds = NULL, initial_shock = NULL,

    tolerance = 1e-07, max_iterations = 500, context = NULL)
```

Arguments

`history`

Long macro release records with variable_id, economic date, release_date and numeric value.

`variable_transforms`

Records containing variable_id, transform identity/log/difference/log_difference, logical standardize and original unit. Fit-only population centers/scales are frozen.

`frequency`

monthly, quarterly or annual; observations and forecast grids must be regular. No implicit interpolation.

`fit_window`

ISO start, end and known_at. Transformation estimation uses only the vintage released by known_at.

`information_cutoff`

ISO cutoff controlling the output vintage, separate from transformation estimation cutoff.

`transformed_history`

MacroHistory or wide dated object: dates, values, variable_ids, per-cell release_dates, units and frequency. Training values/vintage are retained separately when transformed.

`level_history`

Undifferenced level history with the same dated wide schema; integration order is assumed explicitly.

`lags`

Nonnegative observed lags for regimes; positive levels lag count for VAR/VECM. A VECM has lags minus one differenced lags.

`deterministic`

VAR: none, constant, trend or constant_trend. VECM: none, constant_inside, constant_outside, constant_outside_trend_inside or constant_trend_outside. Trend uses original training observation index.

`covariance_estimator`

mle or df_adjusted VAR innovation covariance divisor; no hidden dynamics repair.

`rank`

Declared Johansen cointegration rank from zero to number of variables minus one. No automatic rank or critical-value inference.

`integration_order`

I1. The caller supplies the economic integration assumption; fitting does not establish it.

`training`

start, end, known_at, split_id and optional data_origin. Actual fitted observations determine data_fingerprint. Labels and releases cannot exceed their cutoffs.

`target`

Common target: event, positive horizon, time_unit days/months/years, probability_measure PHYSICAL/RISK_NEUTRAL and conditioning.

`model_id`, `model_version`

Nonempty frozen artifact identifiers.

`context`

Optional common calculation context, which cannot contradict the probability measure or invent a TTC initial law.

`regime_count`

Positive number of joint hidden Gaussian regimes.

`initialization`

Regimes: nonempty list of starts containing transition, initial_probabilities, coefficients and covariances. Coefficients have regime by (intercept plus lag blocks) by response axes; transition rows are origin states. Factors: identified loadings, mean, ar and variances, with factors preceding idiosyncratic states.

`max_iterations`

Positive optimizer/EM iteration cap. Fit diagnostics expose nonconvergence; local optimization never claims global optimality.

`tolerance`

Positive convergence or feasibility tolerance; numerical guards additionally verify support and constraints.

`factor_count`

Positive latent factor count strictly below observed variable count. Loading anchors have unit diagonal and zero upper triangle in the first factor_count rows.

`ar_limit`

Declared absolute stationary AR coefficient bound strictly between zero and one for factor estimation.

`variance_bounds`

Positive increasing lower/upper bounds on factor and idiosyncratic innovation variances, used during estimation.

`residual_history`

Finite residual row matrix, one column per joint macro factor; entire rows are resampled together.

`shared_draws`

Uniform array scenario by time by two: block-start and restart draws in [0,1).

`factor_ids`

Unique ordered macro innovation channel identifiers.

`scenario_ids`

Unique ordered scenario identifiers, shared throughout downstream calculations.

`time_grid`

Consecutive forecast times in years matching the declared frequency.

`block_length`

Moving/circular block integer length; stationary-bootstrap expected block length. Must not exceed available residual history.

`scheme`

moving, circular or stationary multivariate block bootstrap.

`weights`

For bootstrap, optional nonnegative scenario masses. For satellites, a named list mapping exactly training observation IDs to positive precision weights (WLS) or integer frequency weights (GLM); OLS requires unit weights.

`fitted_model`

Portable fitted VAR, VECM, regime or dynamic-factor artifact, or corresponding FitResult. Reverse stress currently requires linear VAR/VECM dynamics.

`innovations`

Named cube with kind STANDARD_NORMAL or RAW_INNOVATIONS, values on scenario/time/channel axes, scenario_ids, factor_ids, time_grid in years and optional weights. Factors use latent state_ids as innovation channels.

`initial_state`

Linear fixed history of shape lags by variables, or scenario-specific histories for a declared cycle mixture. Factors accept a state vector only with DECLARED_FIXED_STATE. Defaults retain fitted terminal state information.

`regime_uniforms`

Scenario by (one initial plus forecast steps) uniform matrix in [0,1), required only for hidden-regime simulation.

`output_space`

model or level. Level inversion uses frozen original-level anchors; alternative fixed/cycle laws require separate explicit anchoring and are rejected by convenience inversion.

`sampling_contract`

Explicit law and assumptions. Linear defaults FORECAST_CONDITIONAL/PIT. STATIONARY_GAUSSIAN needs cycle_law_id, equilibrium_assumptions and initial_state_draws, stable constant dynamics and no trend. DECLARED_CYCLE_MIXTURE needs phase_ids, phase_probabilities matching scenario mass and scenario-specific histories. Factors use FILTERED_GAUSSIAN with initial normal draws, stationary Gaussian, or DECLARED_FIXED_STATE. Zero-lag regimes also permit STATIONARY_MARKOV_EMISSION with a unique invariant regime law.

`conditioning_points`

Records with zero-based time_index, factor_id and exact value. Empty records preserve the unconditional result.

`sampler`

gaussian_equality: full joint Gaussian conditioning with cross-covariance, never isolated overwrites. Empirical bootstrap and regime/cycle mixtures are not silently approximated as Gaussian.

`loss_model`

A function taking a horizon by factor path, or portable kind linear_predictor with same-shaped coefficients, intercept and identity/logit/log link.

`horizon`

Positive number of forecast periods for reverse stress.

`target_loss`

Finite response threshold to reach or exceed.

`shock_metric`

Positive definite precision matrix on flattened time-major standard-normal shocks; objective is half the quadratic form.

`shock_bounds`

Optional per-shock lower/upper bounds for reverse optimization.

`initial_shock`

Optional starting shock vector, one value per horizon/factor coordinate.

`target_series`

Target records: observation_id, entity_id, date, information_date, known_at and value. Only assigned training values are read for estimation.

`macro_history`

Wide dated macro history with ordered variables, units and per-cell release dates. Explicit lagged features must be released at each target information date.

`feature_spec`

Records with feature_id, variable_id, nonnegative lag, physical unit and expected_sign positive/negative/unrestricted. Sign hypotheses are diagnosed rather than silently imposed.

`estimator`

OLS, WLS or Bernoulli GLM. Continuous estimators use native weighted least squares; GLM uses established native binary MLE with rank/separation guards.

`link`

identity/log for continuous response; logit/probit/cloglog for binary GLM. Log inversion without residual draws is a conditional median, not an invented mean adjustment.

`split_plan`

split_id, mode TEMPORAL/ENTITY_TIME, training_cutoff and exhaustive disjoint train_ids plus optional calibration_ids, validation_ids, test_ids. Nonempty holdout follows model availability; entity-time roles also have disjoint entities.

`target_unit`

Explicit unit of the satellite response, e.g. probability, currency or index.

`covariance`

classical, HC1, HAC or cluster coefficient uncertainty. GLM uses classical expected Fisher information. Cluster requires multiple entities; HAC requires one regular entity series.

`hac_lags`

Nonnegative Bartlett/Newey-West covariance lag, smaller than the number of training rows.

`residual_process`

none, white_noise or fitted zero-intercept AR1. Binary GLM requires none. AR1 uses consecutive within-entity pairs and rejects nonstationary estimates.

`fitted_satellite_model`

Frozen macro.fit_satellite_model artifact.

`scenario_cube`

Shared ScenarioCube with axes scenario/time/factor, IDs, normalized weights, units, frequency, year grid, economic origin, information cutoff, measure and full sampling contract.

`idiosyncratic_innovations`

STANDARD_NORMAL object with scenario by time values and exactly matching scenario_ids and time_grid; no hidden independent macro draws.

`initial_target_state`

as_of matching macro origin, known_at covering model/macro availability, scalar residual_state and mode absolute/baseline_delta. macro_history covers exactly the maximum lag ending at origin. Delta mode also supplies baseline_link_path and reference_feature_path.

Details

Transformation fitting uses training-only release vintages and retains an original-level inverse anchor. Later revisions cannot alter frozen training values. VAR reports estimated companion roots without repair. VECM preserves explicit rank, integration and deterministic placement, including rank-zero and one-lag cases. Trace statistics do not automatically infer rank.

The hidden-state EM likelihood uses regime-specific regressions on observed lagged levels. Every supplied start is evaluated and failures retained. Regimes are ordered by intercept, filtered probabilities are distinct from training-window smoothed probabilities, and local convergence is explicit. A one-regime fit reproduces the corresponding Gaussian regression.

Dynamic factors use genuine Gaussian Kalman likelihood with latent and idiosyncratic AR(1) states, identified loading anchors and explicit parameter bounds. Missing cells are integrated out without future imputation. Native R uses KFAS. Frozen artifacts retain observation/transition/covariance matrices and the terminal filtered distribution. Nonconvergence remains visible in fit diagnostics.

Stationary Gaussian TTC initialization solves equilibrium moments only for stable constant dynamics. Unit-root VECMs and deterministic trends fail this law. Declared cycle mixtures verify phase-weight mass but do not establish the caller's economic equilibrium assumptions. Regime stationary probabilities alone do not establish an autoregressive joint invariant law. Factor forecasts include filtered state uncertainty through explicit initial draws. All output cubes preserve the full sampling contract, named axes, measure and information cutoff.

Gaussian equality conditioning updates the entire correlated path and includes latent or stationary initial uncertainty. Support checks use unit-normalized covariance. Singular density is labeled in standardized-support coordinates; full-rank density uses physical coordinates. Density is not probability of an exact continuous event. Conditional stress retains its parent PIT/TTC scope. Reverse optimization reports a local feasible shock or explicit failure; it makes no general global optimality or infeasibility claim.

Satellite fits exclude future releases, immature training labels and held-out target values. WLS covariance conventions distinguish precision weights from binary GLM frequency weights. AR residual pairs do not cross entity boundaries or calendar gaps. Application exposes baseline, macro and idiosyncratic link contributions; zero macro deviations preserve a supplied baseline in delta mode. Shared macro paths and their cycle contract are preserved.

Value

FitResult containing a common portable ModelArtifact for estimators; otherwise CalcResult with a MacroHistory, innovation cube, ScenarioCube or reverse-stress optimization result and diagnostics. Artifact replay uses explicit innovations, not merely matching random seeds.

References

https://www.statsmodels.org/stable/vector_ar.html

https://search.r-project.org/CRAN/refmans/urca/html/ca.jo.html

https://search.r-project.org/CRAN/refmans/KFAS/html/SSModel.html

---

[1.5in]

`market_implied`
Local Market Credit Pricing, Calibration and Historical Physical PD

Description

Eight native methods with explicit risk-neutral cashflows, quoted-population historical physical estimation, and identified and assumed parameter diagnostics.

Usage

```
build_credit_quote_snapshot(quotes, instruments, discount_curve,
    snapshot_contract, context = NULL)

cds_expected_cashflow_legs(hazard_curve, discount_curve,
    cds_contract, context = NULL)

bootstrap_cds_hazard_curve(quotes, discount_curve,
    calibration_contract, context = NULL)

risky_bond_expected_cashflows(hazard_curve, discount_curve,
    bond_contract, context = NULL)

calibrate_bond_hazard_curve(quotes, discount_curve,
    calibration_contract, context = NULL)

diagnose_market_credit_identifiability(calibration,
    alternative_assumptions, diagnostic_contract,
    context = NULL)

fit_market_to_physical_pd_bridge(observations, feature_order,
    bridge_contract, folds, model_id,
    model_version, context = NULL)

compare_cds_bond_credit_basis(cds_calibration, bond_calibration,
    comparison_contract, context = NULL)
```

Arguments

`quotes`

Nonempty dated normalized quote records; spread/upfront fractions or clean prices per 100 par. Calibration rows carry eligible, quote_id, observed_at, known_at and the full instrument; bond fits additionally require price_standard_error.

`instruments`

Nonempty uniquely identified complete CDS/bond contracts with issuer, seniority and currency.

`discount_curve`

Piecewise-constant instantaneous-forward discount contract with as_of, known_at, currency, ACT/365F time_basis, positive increasing knots, aligned rates and explicit FLAT_LAST_RATE extrapolation. Negative discount rates are permitted.

`snapshot_contract`

Daily valuation/knowledge cutoff, currency, nonnegative integer maximum_age_days, stale_policy EXCLUDE or KEEP_FLAGGED and bid_ask_policy REQUIRED or OPTIONAL.

`hazard_curve`

Nonnegative piecewise-constant RISK_NEUTRAL intensity contract using the same dated curve schema as discount_curve.

`cds_contract`

Supplied CDS protection/payment/accrual contract. Required signed upfront and accrued_premium are positive when paid by the protection buyer. Premium mode is CONTINUOUS or SCHEDULED; recovery and annual coupon are explicit.

`calibration_contract`

CDS: fixed recovery, positive maximum_hazard and price_tolerance. Bond: hazard_knots, initial_hazards, maximum_hazard, fixed recovery and liquidity_spread, integer max_evaluations and KNOWN_GAUSSIAN_PRICE_STANDARD_ERRORS weight_interpretation.

`bond_contract`

Supplied coupon/amortization schedule conserving outstanding principal; PAR, TREASURY or MARKET_VALUE recovery, fixed liquidity_spread, accrued_interest and same-day settlement. Optional payment scenarios must be explicitly identified.

`calibration`

Sealed feasible MarketCreditCalibration value returned by a CDS or bond fit.

`alternative_assumptions`

Nonempty records with distinct scenario_id and fixed recovery/liquidity alternatives. Every alternative is actually refitted.

`diagnostic_contract`

Positive quote_standard_errors, nonnegative equivalence_tolerance, recovery_bounds and bond liquidity_bounds; optional dated independent_recovery_observation with standard_error and evidence_id.

`observations`

Dated observed issuer-default records with feature and curve vintages, market_curve_hash, maturity/completeness dates, actual binary outcomes and exact feature mappings. No hypothetical labels.

`feature_order`

Distinct ordered predeclared feature names, exactly matching every observation and market feature contract.

`bridge_contract`

PHYSICAL first-default target, RISK_NEUTRAL feature definitions/horizons, quoted-issuer population, identity transformation, information cutoff, split_id and logit/probit/cloglog link; optional data_origin.

`folds`

Nonempty records with zero-based train and validation arrays. Entity-disjoint forward folds require all training labels before validation dates; warmup predictions stay null.

`model_id`, `model_version`

Nonempty identifiers for the frozen historical physical-PD model.

`cds_calibration`, `bond_calibration`

Sealed feasible same-issuer/seniority CDS and bond calibrations for convention-harmonized comparison.

`comparison_contract`

Common recovery/liquidity assumptions, NO_ADDITIONAL_FUNDING_WEDGE, maximum_quote_age_days and full comparison_cds_contract.

`context`

Optional common calculation context. Market pricing requires RISK_NEUTRAL and the historical bridge requires PHYSICAL when a measure is supplied.

Details

Common contracts:

```
Python contracts are dictionaries with array fields as lists. Native
R uses named
lists for objects and unnamed lists for JSON arrays, including
empty/single arrays.
All date fields are ISO calendar dates (YYYY-MM-DD). The information
clock is
calendar-day resolution; use a separately constructed
cutoff-eligible daily input
when intraday ordering matters. Public functions accept optional
context; its
measure, currency, valuation date and knowledge cutoff must be
compatible where
applicable. Numeric inputs must be finite.

Both hazard_curve and discount_curve contain:

| Field | Meaning |
| --- | --- |
| as_of, known_at | Valuation origin and input vintage; knowledge
cannot follow valuation. |
| currency | Explicit matching currency identifier. |
| time_basis | Exactly ACT/365F: actual calendar days divided by
365. |
| knots | Strictly increasing positive interval ends in years from
origin. |
```

```
| rates | One rate per interval; final rate continues after the
final knot. |
| extrapolation | Exactly FLAT_LAST_RATE, explicitly authorizing
that continuation. |
| probability_measure | Required RISK_NEUTRAL for hazard curves. |

Hazards are nonnegative. Discount forward rates may be negative.
Discount factors
are D(t)=exp(-integral r) and survival is S(t)=exp(-integral
lambda).
Very large exponents fail explicitly instead of overflowing or
silently producing
zero survival. Curve fitting records the latest quote/discount input
knowledge
in the fitted hazard vintage; an earlier known discount curve can
price a later
eligible quote at the same valuation origin.

Instrument contracts share as_of, currency,
probability_measure=RISK_NEUTRAL
and default_settlement=AT_DEFAULT. Calibration additionally requires
instrument_type (CDS or BOND), issuer_id and seniority on every
instrument.
Each calibration fits a single issuer and seniority.
```

Local quote snapshot:

```
build_credit_quote_snapshot(quotes, instruments, discount_curve,
snapshot_contract, context=None)

instruments is a nonempty array with unique instrument_id and
complete
instrument contracts. Each quote has unique quote_id, instrument_id,
quote_type, unit, value, observed_at, known_at, and optional paired
bid, ask. Knowledge cannot precede observation. Supported
conversions are:

| quote_type | Accepted units | Normalized unit |
| --- | --- | --- |
| CDS_SPREAD | BASIS_POINTS, FRACTION | Annual coupon fraction |
| CDS_UPFRONT | PERCENT_NOTIONAL, FRACTION | Fraction of notional |
| BOND_CLEAN_PRICE | PER_100_PAR | Price per 100 outstanding par |

snapshot_contract requires as_of, known_at, currency, integer
maximum_age_days >= 0, stale_policy (EXCLUDE, KEEP_FLAGGED) and
bid_ask_policy (REQUIRED, OPTIONAL). It returns all normalized rows
with
eligible and findings. Future information, currency mismatch,
crossed/outside
```

```
bid-ask quotes and conflicting values for the same
instrument/type/vintage are
ineligible. Stale rows are either excluded or explicitly retained
with a flag.
Different seniorities remain separate and appear in
cross_seniority_groups.
The snapshot is canonically hashed. It never downloads market data.

The calibration quotes argument accepts eligible normalized snapshot
rows
(snapshot.value['quotes'] filtered by eligible) or equally explicit
local
rows. Each needs quote_id, eligible=True, observed_at, known_at,
instrument, quote_type, normalized_unit, value. Quote knowledge must
not
exceed valuation. Eligibility assertions supplied without a snapshot
remain the
caller's data-preparation contract; calibration cannot attest to
unseen quotes.
```

CDS cashflows and bootstrap:

```
cds_expected_cashflow_legs(hazard_curve, discount_curve,
cds_contract, context=None)

cds_contract additionally requires recovery in [0,1], nonnegative
annual
coupon, protection_start, maturity, settlement_date, signed
accrued_premium, signed upfront, and premium_mode. Positive upfront
and
settlement-accrued amounts are paid by the protection buyer;
negative amounts are
rebates received by the buyer. All results are per unit notional.

* CONTINUOUS: premium integrates D(t) S(t) over the protection
interval.
  Settlement accrued premium must be zero. This provides the exact
flat oracle
  par_spread=(1-recovery)*hazard.
* SCHEDULED: requires day_count (ACT/360 or ACT/365F), boolean
  accrual_on_default, and contiguous schedule rows with
accrual_start,
  accrual_end, payment_date. Accrual end cannot exceed maturity; the
last end
  equals maturity. The first period can start before valuation.
Coupon payment
  can lag accrual end: survival is tested at accrual end and the
surviving coupon
  is then owed at payment, with no additional post-maturity default
```

```
risk.
  Default accrual integrates actual linear accrued fraction,
including any
  pre-valuation portion, over the covered default interval.

Protection PV is (1-R) integral D S lambda. The premium and
accrued-default
annuities are separate.
fair_upfront=(protection-coupon*risky_annuity)/D(settlement)
-accrued_premium; the residual discounts the difference from
supplied upfront.
par_spread excludes settlement accrued cash, so it is the ratio of
protection
PV to total risky annuity, not an implicit rescaling of a supplied
accrued quote.
The payment/accrual conventions are deliberately explicit.

bootstrap_cds_hazard_curve(quotes, discount_curve,
calibration_contract, context=None)

The contract requires fixed recovery, positive maximum_hazard and
positive
price_tolerance. Maturities must be positive and distinct. Spread
quotes set
coupon to the normalized spread and upfront to zero, and require
zero settlement
accrued premium. Upfront quotes retain the instrument's fixed coupon
and accrued
settlement amount. A bounded scalar root is solved for each next
hazard segment.

The returned sealed MarketCreditCalibration includes feasible, the
fitted or
partial hazard_curve, quote residuals, survival, solver bracket
residuals and
boundary statuses. A missing nonnegative bracket or unmet residual
tolerance is
reported; negative hazards are never clipped. Partial curves are not
accepted by
basis/identifiability methods. Local knot sensitivities identify
their fixed-prior-
knots scope; curve_sensitivities additionally inverts the full
quote/hazard
Jacobian for recursive quote and recovery effects. Full
sensitivities are withheld
at a parameter boundary or deficient rank. A perfectly fitted zero
spread with
100% recovery does not identify a hazard.
```

Bonds and hazard calibration:

```
risky_bond_expected_cashflows(hazard_curve, discount_curve,
bond_contract, context=None)

The contract requires recovery, fixed liquidity_spread,
recovery_convention (PAR, TREASURY, MARKET_VALUE),
cashflow_assumption (CONTRACTUAL_NONOPTIONAL,
EXOGENOUS_OPTION_SCENARIO),
positive outstanding_principal, nonnegative accrued_interest,
settlement_date=as_of and nonempty schedule. Each schedule row has
strictly
increasing future payment_date, nonnegative coupon_amount,
principal_amount.
Principal payments must sum to outstanding principal. Optional
cashflows require
an additional option_scenario_id; an externally supplied scenario is
not an
option price. Bond prices are at same-day settlement, with no hidden
funding or
forward-settlement adjustment.

* PAR: recovery at default is R times remaining outstanding
principal. The
  principal decreases only at scheduled amortization dates.
* TREASURY: recovery is R times the then risk-free value of all
remaining
  promised coupon/principal payments. This differs from recovery of
par.
* MARKET_VALUE: recovery is R times pre-default value. The
deterministic
  valuation discounts promised cashflows at r + liquidity +
(1-R)*lambda.

Survival-weighted promised coupon and principal PVs remain separate
from recovery
PV under all conventions. For market-value recovery, recovery PV is
the exact
residual between the reduced-form value and survival-weighted
promised payments.
Liquidity is an exogenous discount wedge applied to both promised
and default
cashflows. Treasury's replacement claim uses the risk-free curve
without that
wedge. Output separately reports the risk-free, credit-only and
credit-plus-
liquidity prices and their additive price effects. Clean price is
dirty price
minus supplied accrued interest; clean_price_per_100 divides by
```

```
outstanding par.
No credit/liquidity attribution is identified merely by this
arithmetic.

calibrate_bond_hazard_curve(quotes, discount_curve,
calibration_contract, context=None)

Each bond quote additionally requires strictly positive
price_standard_error
in price-per-100 units. The contract requires hazard_knots,
initial_hazards,
positive maximum_hazard, fixed recovery, fixed liquidity_spread,
positive
integer max_evaluations, and
weight_interpretation=KNOWN_GAUSSIAN_PRICE_STANDARD_ERRORS. Native
bounded
nonlinear optimization minimizes the sum of squared price residuals
divided by
those known errors. Python uses bounded least squares; R uses native
L-BFGS-B
with a numerical pricing Jacobian. Solver iteration counts need not
match.

Outputs retain raw/weighted residuals, fitted prices, objective,
convergence,
Jacobian, column-scaled rank/condition, active boundaries and
covariance when
available. Intervals are untruncated local 95% Wald approximations
conditional
on fixed recovery/liquidity, model correctness and supplied Gaussian
price errors.
They are withheld at boundaries or deficient rank; they are neither
global
profile intervals nor evidence that recovery or liquidity was
learned from data.
Overparameterized fits remain flagged instead of claiming unique
identification.
```

Identifiability and basis:

```
diagnose_market_credit_identifiability(calibration,
alternative_assumptions, diagnostic_contract, context=None)

calibration is a sealed feasible CDS/bond calibration value.
alternative_assumptions is a nonempty array with unique scenario_id,
recovery
and, for bonds, liquidity_spread. Every alternative actually refits
hazards.
diagnostic_contract requires nonnegative equivalence_tolerance, one
```

```
positive
quote_standard_errors value per quote, two-element recovery_bounds,
and for
bonds two-element liquidity_bounds. Bounds must contain the
reference and all
alternatives. The objective is squared standardized quote residuals,
optionally
plus independent recovery information.

Optional independent_recovery_observation requires value, positive
standard_error, nonempty evidence_id, and eligible known_at. Its
independent
Gaussian measurement assumption is explicit. The joint Jacobian
includes hazard,
recovery and (bond) liquidity columns plus the independent
observation row.
Profiles report actual alternative hazards/objectives, failed
feasibility and
near-equivalent sets. This is a finite declared scenario profile,
not a search
proving the global identified set. The normal equivalence tolerance
is a supplied
objective tolerance, not an automatically inferred confidence
threshold.

compare_cds_bond_credit_basis(cds_calibration, bond_calibration,
comparison_contract, context=None)

The comparison contract requires fixed recovery,
bond_liquidity_spread,
funding_treatment=NO_ADDITIONAL_FUNDING_WEDGE, integer
maximum_quote_age_days, and a full comparison_cds_contract. The
method checks
issuer, seniority, currency, curve valuation/vintage, exact discount
conventions,
recovery, liquidity, quote freshness and maturity coverage.
Mismatches return
comparable=False, explicit findings and null basis, rather than an
unmarked
number. Matching curves are repriced with the same CDS schedule. The
sign is
CDS par spread minus bond-implied equivalent CDS par spread. Output
includes both
spreads, basis in fractions/basis points and survival difference. A
matched
synthetic world gives zero basis. This is a convention-conditional
market
comparison, not a pure default signal or trading recommendation.
```

Historical physical-PD bridge:

```
fit_market_to_physical_pd_bridge(observations, feature_order,
bridge_contract, folds, *, model_id, model_version, context=None)

Each observation contains unique observation_id, entity_id, as_of,
label_known_at, complete_through, market_curve_as_of,
market_curve_known_at,
market_curve_hash, features_known_at, features mapping exactly the
declared
feature names, binary outcome, default_date (null for no observed
default),
initial_state=PERFORMING, and
outcome_origin=OBSERVED_ISSUER_DEFAULT.
Market features/curve vintages must be available by the prediction
date; target
windows must be fully mature and available by fit cutoff. A default
must occur
within the declared target horizon. Feature hashes document supplied
lineage;
the method does not reconstruct unseen historical quote books.

The bridge contract requires:

* target: event=first_default, positive horizon, time_unit in
days/months/
  years, probability_measure=PHYSICAL,
conditioning=initially_performing.
  Date windows use integer days or an integer number of calendar
months, clamping
  calendar month ends.
* feature_probability_measure=RISK_NEUTRAL,
market_feature_contracts: one
  object per feature with probability_measure=RISK_NEUTRAL, positive
horizon,
  time_unit and nonempty definition. Feature horizons may differ
from physical
  target horizons because the learned conditional model is explicit.
* outcome_origin=OBSERVED_ISSUER_DEFAULT, nonempty
target_population,
  population_selection=MARKET_QUOTED_ISSUERS,
  transformation=IDENTITY_PREDECLARED_FEATURES, known_at, split_id,
  link (logit, probit, cloglog), and optional data_origin.

folds contains nonempty unique zero-based train and validation index
arrays.
Training and validation entities are disjoint, validation rows occur
in at most
one fold, and every training label is known strictly before the
```

earliest
validation prediction date. Each fold fits an actual Bernoulli GLM;
no full-fit
predictions are substituted. Warmup rows have null out-of-time
predictions.
The final model fits all eligible observed rows for subsequent
prediction.
Out-of-time Brier/log loss and entity-cluster Brier uncertainty use
only real
observed outcomes. Native GLM coefficient covariance is also
retained. Separation,
rank deficiency and nonconvergence fail explicitly.

The frozen model uses common ModelArtifact serialization and generic
predict_pd dispatch, requiring matching features and the PHYSICAL
target. An
estimated conditional historical association is not a universal
Radon-Nikodym
measure change, a causal premium decomposition, an
unquoted-population estimate,
or a justified TTC estimate. Selection into the quoted population
remains an
explicit limitation. There is no automatic identity Q-to-P
conversion.

References and checks: The recovery distinctions and reduced-form valuation follow the definitions in Darrell Duffie, Defaultable Term Structure Models with Fractional Recovery of Par (1998). ISDA CDS Standard Model documentation illustrates the broader contractual conventions necessary for standard-model reproduction; this implementation makes no such equivalence claim.

Default tests include actual fits, independent closed-form prices, independent accrual quadrature, amortization conservation, inverse quote calibration, recursive sensitivity refits, nonidentification with and without independent recovery data, zero-basis matching, temporal/entity leakage and frozen model interchange. CREDITRATINGSYSTEM_EXTENDED_TESTS=true adds the recovery/liquidity/recovery- convention calibration grid. Default CI avoids that exhaustive grid.

Value

A CalcResult with pricing, quote snapshot, calibration, identifiability or basis values; the historical bridge returns FitResult with a common ModelArtifact. Infeasible CDS calibrations retain explicit partial results. The details describe all result fields and limitations.

See Also

predict_pd, serialize_fitted_model, load_fitted_model

[1.5in]

monitoring
Dated Local Model Roles and Monitoring Escalation

Description

Preserve model-role event history and apply explicit monitoring policies without external deployment or recalibration.

Usage

```
assign_model_role(history, registry, assignment, context = NULL)
rollback_model_assignment(history, registry, target_event_id,
  rollback_event, context = NULL)
classify_monitoring_findings(metric_panel, policy, context = NULL)
```

Arguments

`history`

NULL initially, or a ModelAssignmentHistory value or CalcResult.

`registry`

An immutable local rating model registry.

`assignment`

Named event object with event_id, profile_id, purpose, role, slot_id, model_id, model_version, model_hash, effective_from, known_at, evaluation_reference, reason and required_feature_order. Arrays are unnamed lists, including an empty feature array.

`target_event_id`

Explicit historical assignment event identifying the rollback model and scope.

`rollback_event`

New event_id, effective_from, known_at, reason, evaluation_reference and required_feature_order. Other model/scope fields cannot contradict the target.

`metric_panel`

Rows with series_id, metric, period, status and value (possibly NULL). One row per series/metric/period.

`policy`

Schema1.0 named object with policy_id, version, periods, series_ids and rules. Periods and series IDs are distinct nonempty arrays. Each rule has rule_id, metric, operator, threshold, consecutive_periods, severity and action_code.

`context`

Optional calculation context.

Details

Roles are champion or challenger. A profile/purpose has one champion slot named champion; named challenger slots allow parallel evaluations. Events append in nondecreasing knowledge order. Repeating identical content is idempotent; conflicting IDs or equal role/slot effective boundaries fail. Registered model content and exact feature order must match. References are user assertions.

`select_effective_model` accepts the history and reconstructs champion intervals using only events known at its cutoff. Intervals include the start and exclude the next known event's start. Events remain unchanged; a backdated later-known assignment cannot affect an earlier knowledge replay. History alone has no final expiry. Rollback is a new event and does not cancel independently scheduled future changes.

Monitoring operators are gt, ge, lt or le. Persistence is a positive integer; severity is WARNING or CRITICAL. Every expected series is required, so an entire absent series is insufficient information. Supported input statuses are OK, MISSING_OUTCOME, INSUFFICIENT_INFORMATION, TECHNICAL_ERROR and NOT_APPLICABLE. Missing information breaks the breach streak. A premature breach is WATCH; a known nonbreach is WITHIN_POLICY. The trace retains every period; overall status describes only the last declared period. Precedence is technical error, critical, warning, missing outcome, insufficient information, not applicable, watch, within policy. Only persisted warning/critical breaches receive the supplied action code.

Value

A CalcResult containing a content-hashed append-only assignment history or a monitoring status with full rule trace.

[1.5in]

`monitoring_metrics`
Mature-outcome monitoring, distribution drift and paired diagnostics

---

Description

Local quantitative monitoring under explicit coverage, maturity, cluster resampling and null-path contracts. No model promotion or recalibration is performed.

Usage

```
build_monitoring_cohorts(predictions, outcome_events, coverage,
  target_contract, evaluation_cutoff, context = NULL)
monitor_rating_cohort(cohort, group_by, reference_window,
    minimum_count,
  uncertainty, cluster_resamples = NULL, confidence_level = .95,
  context = NULL)
detect_credit_feature_drift(reference, current, feature_order,
    method,
  permutations, multiple_testing, dependence_assumption,
  reference_weights = NULL, current_weights = NULL,
```

```
    bandwidth = NULL,
  frozen_bins = NULL, smoothing = .5, permutation_units = NULL,
  context = NULL)
detect_pd_calibration_drift(residuals, method, parameters,
    null_paths,
  null_spec, alpha, validation_null_paths = NULL,
  confidence_level = .95, context = NULL)
compare_challenger_cohorts(champion, challenger, cluster_resamples,
  selection_protocol, confidence_level = .95, context = NULL)
diagnose_fairness_slices(cohort, group_labels, reference_group,
  use_basis, minimum_count, risk_threshold, cluster_resamples,
  confidence_level = .95, context = NULL)
```

Arguments

`predictions`

List of records: forecast_id, case_id, entity_id, cluster_id, cohort, segment, model_id, model_version, forecast_at, known_at, pd and exact target contract. Optional weight, origin_grade, override, decision and prediction_origin.

`outcome_events`

Records with unique event_id, entity_id, event_at, known_at, default_definition and event (default, exit or rating). A rating also requires grade.

`coverage`

Records with entity_id, complete_through and known_at at or after completion, declaring complete event ascertainment.

`target_contract`

Named list: event=default, probability_measure=PHYSICAL, default_definition, unit, positive integer horizon_days, optional default_grade. Entity identifiers denote the declared unit.

`evaluation_cutoff`

ISO date; later-known outcomes and coverage are excluded.

`cohort`, `champion`, `challenger`

Builder CalcResult or unchanged content-hashed cohort bundle.

`group_by`

Unique grouping fields from cohort rows.

`reference_window`

Named start/end ISO forecast dates, inclusive, applied within each group.

`minimum_count`

Positive integer support threshold.

`uncertainty`

none, iid_homogeneous_binomial, or independent_cluster_bootstrap.

`cluster_resamples`

At least two supplied lists/vectors of cluster identifiers. Every path samples the original number of distinct clusters with replacement. All rows of selected clusters remain together.

`confidence_level`

Interior probability for pointwise percentile or exact rate intervals.

`reference`, `current`

Nonempty lists of feature records.

`feature_order`

Unique feature names.

`method`

Feature drift: ks, wasserstein, mmd_rbf or psi. Sequential drift: cusum or ewma.

`permutations`

B by pooled-N numeric matrix of zero-based complete permutations, even in R. At least two rows; NULL for PSI.

`multiple_testing`

none, holm, bonferroni; not_applicable for descriptive PSI.

`dependence_assumption`

exchangeable_rows or exchangeable_clusters for inference; descriptive for PSI.

`reference_weights`, `current_weights`

Finite nonnegative weights with positive sums; default unit weights.

`bandwidth`

Positive prespecified RBF kernel bandwidth.

`frozen_bins`

Per-feature named specifications: numeric kind, increasing cutpoints and two support endpoints; or categorical kind and unique string levels.

`smoothing`

Positive pseudo-count per weighted PSI bin.

`permutation_units`

Pooled cluster labels for intact equal-size, internally ordered cluster permutation.

`residuals`

Strictly ordered records with ISO time, finite centered value, mature=TRUE. The caller derives counts, rates or standardized residuals under the declared scale.

`parameters`

CUSUM allowance at least zero; or EWMA smoothing in (0,1].

`null_paths`

B by T finite matrix of independent complete null simulation paths matching the monitored horizon. Within-path dependence must reflect the null model.

`null_spec`

model_id, model_version, calibration_id, residual_scale, probability_measure=PHYSICAL, dependence (independent or prespecified_dependent), calibrated_before no later than monitoring start. Independent validation additionally requires validation_paths_independent=TRUE.

`alpha`

Interior target finite-horizon null crossing probability.

`validation_null_paths`

Optional independent C by T paths for empirical false-alarm assessment with exact binomial interval.

`selection_protocol`

protocol_id, evaluation_role=OUT_OF_TIME_TEST, and multiple_challenger_policy single_prespecified or separate_selection_and_final_test. Paired predictions require prediction_origin=OUT_OF_TIME.

`group_labels`

Unique case_id/group records; labels are supplied, never inferred.

`reference_group`

Observed group used for paired differences.

`use_basis`

Nonempty documented permission/purpose for group analysis.

`risk_threshold`

Positive prediction is PD greater than or equal to this probability.

`context`

Optional calculation context.

Details

The default window is open at forecast time and closed at forecast plus horizon_days. Zero labels require mature, fully covered windows. Early known defaults remain positive but ineligible before horizon maturity. Prior defaults/exits exclude forecasts; later exits censor unobserved outcomes. Cure never removes the first default within the event horizon. LGD and incomplete workout labels are not treated as binary default observations.

Migration grades derive from dated, known rating events; prediction outcome_grade is ignored. Origin grade carries forward only under complete mature coverage. Coverage used for migration must also certify complete rating-event capture. First default uses the declared default grade. Hashes detect bundle edits but do not prove source declarations.

Cohort metrics use weighted AUC/Gini, Brier, exact log loss and calibration errors. One-class AUC is null. Impossible log loss retains infinite status and null numerical loss. Exact Clopper–Pearson rate intervals require unit weights, unique clusters and independent homogeneous Bernoulli outcomes. Bootstrap quantiles use type 7; unidentified replicate counts and degenerate bands remain visible.

KS is a weighted empirical CDF supremum. Wasserstein integrates the absolute CDF difference. MMD is the squared weighted empirical RBF embedding distance including diagonals, not the unbiased U-statistic. Missing values are complete-case exclusions for numeric tests and distinct PSI bins. PSI uses smoothed fixed-bin shares and separate unknown-category bins; it has no universal significance threshold.

Numeric permutation p-values equal (1 + null exceedance count)/(B+1), using a relative 1e-14 comparison tolerance. The reported Monte Carlo standard error is approximate and requires independent uniform valid permutations. A 95 percent exact binomial interval for the underlying permutation-tail probability remains nondegenerate at sample boundaries. Hand-selected paths check arithmetic only; exhaustive enumeration inference is not claimed. Cluster paths must move complete equal-size clusters, preserving within-cluster order; weights travel with observations. Marginal multiplicity correction is explicitly chosen. Input drift does not establish concept drift.

CUSUM initializes two sums at zero and updates max(0, up+residual-allowance) and max(0, down-residual-allowance); statistic is their maximum. EWMA initializes at zero, updates smoothing*residual+(1-smoothing)*previous, and uses the absolute value. Threshold is ascending null-path-maximum order statistic ceiling((B+1)*(1-alpha)). Insufficient simulation count is rejected. Crossing is strictly greater than threshold. Exchangeability of calibration maxima and a future whole null path yields an unconditional finite-horizon rank bound; it does not guarantee performance under a misspecified or estimated null. Calibration exceedance frequency and optional independent validation are distinct. Horizon extension requires recalibration.

Challenger differences are challenger minus champion on jointly eligible matching cases. Missing counterparts and eligibility disagreements are separate outputs. Group selection rates use all known decisions, while outcome errors, TPR/FPR and calibration use mature observed cases. Fairness gaps use shared cluster paths; intervals are pointwise. Selective outcomes, small groups and boundary samples limit interpretation. No automatic deployment or fairness certification is implied.

Value

A CalcResult with the cohort bundle, monitoring panel, feature findings, sequential path and calibrated threshold, paired metric differences, or group slices, plus diagnostics and explicit limitations.

References

Gretton et al. (2012), A Kernel Two-Sample Test, Section 2.2, equation (5): https://www.jmlr.org/papers/volume13/gretton12a/gretton12a.pdf. NIST CUSUM and EWMA recurrences: https://www.itl.nist.gov/div898/handbook/pmc/section3/pmc323.htm and https://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm. Kleinberg, Mullainathan and Raghavan (2017), Inherent Trade-Offs in the Fair Determination of Risk Scores: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2017.43.

[1.5in]

pd_term_structure_extended
Default curve inversion, annual hazards and rating migration

Description

Separates conditional, cumulative and marginal default probabilities, and discrete migration from continuous-time intensity models.

Usage

```
conditional_pd_from_cumulative(cumulative_pd, context = NULL)
hazard_to_pd(hazard, interval_lengths, context = NULL)
estimate_transition_matrix(ratings, observation_windows, states,
    default_state,
  estimator, knowledge_date, observation_scheme, exits = NULL,
  cohort_period_days = NULL, days_per_year = 365.25,
  empty_row_policy = "error", context = NULL)
project_transition_matrix(matrix, horizon, matrix_kind, states,
    default_state,
  step_length = NULL, time_unit = "years", context = NULL)
```

Arguments

`cumulative_pd`

Nondecreasing finite probability vector.

`hazard`

Nonnegative annual hazard vector.

`interval_lengths`

Positive year fractions, one per hazard.

`ratings`

Record list or data frame with entity_id, at, known_at, state. Dates are ISO calendar dates.

`observation_windows`

Records entity_id, start, end, complete_until, known_at. One window per entity; completeness must be known by cutoff.

`states`

Ordered unique state labels.

`default_state`

Absorbing default label among states.

`estimator`

cohort or duration.

`knowledge_date`

ISO date information cutoff.

`observation_scheme`

exact_event_times or state_carried_until_change. Duration requires exact_event_times.

`exits`

Optional records entity_id, at, known_at. Exits censor follow-up; same-day rating changes are included before censoring.

`cohort_period_days`

Required positive integer common window duration for cohort estimation.

`days_per_year`

Positive calendar-day conversion basis for annual exposure and cohort step length.

`empty_row_policy`

error, or explicit identity assumption for unidentified nondefault rows.

`matrix`

Square origin-row/destination-column transition matrix or annual generator.

`horizon`

Nonnegative projection horizon in years.

`matrix_kind`

transition or generator.

`step_length`

Positive years per discrete transition matrix; absent for generator.

`time_unit`

years.

`context`

Optional consistent calculation context.

Details

Conditional periods after complete default have NULL values and an undefined mask. Annual hazards integrate over positive year fractions using stable exponential arithmetic.

Cohort estimates divide complete-case endpoint counts by origin counts. Censored records are excluded unless default was already observed; informative censoring can bias this estimator. Duration estimates divide all observed off-diagonal event counts by originating exposure time and set diagonal intensities to negative row totals. This is a homogeneous Markov maximum-likelihood estimator only when all actual state-change times are observed. Periodic panel observations require a different likelihood. State at window start is the last known rating at or before start; later events occur on (start,end]. Future-known records are ignored; duplicate rating dates require upstream resolution. Default is absorbing and observed cures are rejected. Unknown rows error unless no migration is explicitly assumed and reported.

Discrete projection uses an integer number of periods and repeated matrix multiplication. Generator projection uses native Matrix matrix exponential. Inputs must have valid stochastic or generator rows and exactly absorbing default. Projection assumes time-homogeneous Markov dynamics; no generator is inferred from a transition matrix.

Value

CalcResult. Curve inversion returns conditional_pd and undefined_after_full_default. Hazard conversion returns conditional_pd, cumulative_pd, survival and marginal_pd. Estimation returns matrix, matrix_kind, counts and exposure_years, with exclusions and unidentified rows in diagnostics. Projection returns matrix, pd_by_origin and survival_by_origin.

References

Jackson (2011), Multi-State Models for Panel Data: The msm Package for R, doi:10.18637/jss.v038.i08. See also `Matrix::expm`.

[1.5in]

`pit_ttc_transform`
Explicit Gaussian factor PIT and TTC transformation

Description

Conditions or unconditions a latent Gaussian default threshold under an explicit factor model. Scalar inputs expand to equal-length vectors.

Usage

```
pit_ttc_transform(pd, systematic_factor, asset_correlation,
    direction,
  factor_distribution, horizon, time_unit, probability_measure,
    context = NULL)
```

Arguments

`pd`

Physical default probabilities in [0,1].

`systematic_factor`

Finite standard-normal factor values; positive means good conditions.

`asset_correlation`

Asset correlation in [0,1).

`direction`

Either ttc_to_pit or pit_to_ttc.

`factor_distribution`

Named list with family standard_normal, factor_id, reference_period, model_version.

`horizon`

Positive common horizon.

`time_unit`

days, months or years.

`probability_measure`

PHYSICAL.

`context`

Optional calculation context.

Details

PIT equals Phi((Phi-inverse(TTC)-sqrt(rho)*z)/sqrt(1-rho)). TTC integrates over the factor distribution; a zero factor does not generally yield TTC. Correlation and factor suitability are user assumptions. Boundary saturation can lose inverse information and is reported.

Value

CalcResult containing probabilities and factor, horizon and conditioning diagnostics.

References

BIS (2005), An Explanatory Note on the Basel II IRB Risk Weight Functions, https://www.bis.org/publications/explanatory-note-basel-ii-irb-risk-weight-functions.

[1.5in]

`portfolio_simulation`
Shared Default Groups, Portfolio Migration and Paired Dependence Stress

Description

Native portfolio preparation, factor or supplied-copula default paths, once-only terminal guarantee protection, physical rating migrations and paired stress comparisons. Defaults are shared across all positions in a declared default-equivalence group.

Usage

```
prepare_credit_portfolio(positions, entity_graph,
    default_link_policy,
  portfolio_contract, guarantee_allocations = NULL,
  netting_contracts = NULL, context = NULL)
simulate_credit_portfolio(prepared_portfolio, group_pds,
    dependence_contract,
  shared_innovations, conditional_loss_paths, levels = c(.95, .99),
  context = NULL)
simulate_portfolio_migrations(initial_grades, transition_matrices,
  shared_uniforms, migration_contract, valuation_contract = NULL,
  context = NULL)
stress_portfolio_factor_structure(prepared_portfolio, group_pds,
  dependence_contract, shared_innovations, conditional_loss_paths,
  stress_cases, levels = c(.95, .99), context = NULL)
```

Arguments

`positions`

List of exposure-lot records: position_id, claim_id, facility_id, entity_id, currency, ead, facility_ead, sector, region, exposure_basis GROSS/NETTED, collateral_treatment NONE/ALREADY_IN_CONDITIONAL_LGD and optional netting_contract_id. Position and claim IDs are unique; all lots of a facility must conserve its common facility_ead.

`entity_graph`

Sealed relationship graph from build_entity_graph, with the same as_of and known_at cutoffs as the portfolio.

`default_link_policy`

Explicit default-equivalence policy consumed by assign_default_groups. No member PD is averaged or copied. Unresolved equivalence conflicts cannot be simulated.

`portfolio_contract`

Plain JSON-compatible list containing portfolio_id, version, currency, ISO as_of/known_at, positive horizon, time_unit days/months/years, probability_measure PHYSICAL, conditioning initially_performing, event first_default. Guarantee use additionally requires guarantee_

loss_rule FIRST_LOSS_CAP_IF_GUARANTOR_SURVIVES_HORIZON or PRO_RATA_EAD_CAPPED_IF_GUARANTOR_SURVIVES_HORIZON. Calendar horizons use end-of-month clamping.

`guarantee_allocations`

Optional list of sealed simultaneous allocations from allocate_guarantee_coverage. Allocation cut-offs/currency and terminal claim due dates must match. Each guarantee capacity is consumed in only one allocation artifact. Claim owner and guarantor default group must be known.

`netting_contracts`

Optional records containing netting_contract_id, facility_ids, currency, legally_effective TRUE, evidence_reference, valid_from, valid_until and known_at. Netted EAD is already net of the evidenced agreement; this routine performs no second offset. Unused contracts are rejected.

`prepared_portfolio`

Sealed output of prepare_credit_portfolio, or its CalcResult. Tampered positions, protection or target metadata fail the portfolio hash.

`group_pds`

Named list with exactly one probability in [0,1] per frozen default_group_id. These are physical first-default probabilities at the explicit portfolio horizon.

`dependence_contract`

Plain JSON-compatible list with kind and ordered group_ids. SUPPLIED_DEPENDENT_UNIFORMS consumes group_uniforms. GAUSSIAN_FACTORS additionally declares ordered factor_ids, factor_correlation and group-by-factor loadings. The matrix is validated without substantive repair and systematic latent variance cannot exceed one.

`shared_innovations`

Plain JSON-compatible list with ordered scenario_ids, group_ids, weights and sampling_design IID/FIXED_WEIGHTED_SCENARIOS/DEPENDENT_OR_QMC/EXACT_STATES. Supply scenario-by-group group_uniforms in [0,1], or scenario-by-factor factor_normals and scenario-by-group group_normals. Nested arrays are explicit lists for portable hashing. Gaussian common/residual coordinates must be independent standard normals under the declared sampling law.

`conditional_loss_paths`

Plain JSON-compatible list binding ordered scenario_ids and position_ids to driver_hash of the dependence contract plus shared innovations, scenario-by-position ead and raw lgd, and guarantee_treatment NOT_YET_APPLIED. EAD is nonnegative; signed or above-one conditional LGD is retained. Arrays use explicit nested lists for portable hashing.

`levels`

Distinct VaR/expected-shortfall levels in [0,1]. Discrete boundary mass and level-one conventions follow credit_loss_distribution.

`initial_grades`

Named list with one known state for every migration group.

`transition_matrices`

Square homogeneous physical transition matrix, or period-by-origin-state-by-destination-state numeric array/list according to matrix_mode. Rows must sum to one within numerical tolerance; only that rounding normalization is applied.

`shared_uniforms`

Plain JSON-compatible list containing scenario_ids, group_ids, scenario-by-period-by-group values in [0,1), and nonnegative weights. Draws must satisfy the declared sequential conditional-uniform law. Same-period group draws may be correlated. Nested arrays are explicit lists for portable hashing.

`migration_contract`

contract_id, version, ordered group_ids and states, default_state, logical allow_cure, probability_measure PHYSICAL, increasing positive time_grid, time_unit, matrix_mode HOMOGENEOUS/TIME_VARYING and innovation_law CONDITIONALLY_UNIFORM_GIVEN_PRIOR_STATES. Default is absorbing unless cure is explicitly allowed.

`valuation_contract`

Optional plain list with valuation_id, version, currency, valuation_basis SUPPLIED_FROZEN_MARKED_VALUES, matching group_ids/states, group initial_values, group-by-state terminal_values_by_grade and scenario-by-group cash_distributions. Physical migration probabilities are not converted into pricing probabilities.

`stress_cases`

List of distinct case_id records with dependence_contract, group_pds, conditional_loss_paths and recovery_response FIXED_PATHS/REPLACED_CONDITIONAL_PATHS. Fixed paths require unchanged EAD/LGD and identities. Every case reuses the same supplied random coordinates and has a recomputed driver_hash.

`context`

Optional common calculation context consistent with currency, horizon, time unit, measure and conditioning where applicable.

Details

The prepared artifact binds facility lots, default groups and reservations with a content hash. Complete legal evidence, facility totals and default-link policy are supplied by the caller; they are not legally authenticated. Collateral recovery belongs in the conditional LGD path and is not reapplied.

For Gaussian factors, common normals are transformed by the declared PSD correlation matrix; systematic loadings plus independent residual normals produce unit-variance latent assets. Asset correlation is distinct from default-indicator correlation. Supplied dependent uniforms can instead come from the granular copula methods. Each group produces one default event that applies to all its facilities. EAD and LGD paths are bound to the same scenario/driver contract, preserving supplied dependence with recovery and exposure.

Gross loss is default indicator times EAD times raw LGD. Guarantees pay only positive remaining loss when the borrower defaults and the guarantor survives the terminal horizon. First-loss protection pays up to its reserved monetary cap. Pro-rata protection first scales positive gross loss by

the covered-EAD fraction capped at one, then applies monetary and remaining-loss caps. These are explicit terminal-horizon payoff assumptions, not a regulatory capital substitution formula or a model of interim guarantee settlement.

Migration paths retain the initial grade, every period grade and first-default time. First-default time remains fixed after an explicitly allowed cure. The output separately reports weighted empirical terminal frequencies and products of the declared transition matrices. Arbitrary temporal dependence of supplied uniforms does not establish the Markov law. Optional marked loss equals supplied initial value minus supplied terminal mark minus cash distributions; signed valuation gains remain present.

Dependence stress preserves random coordinates and labels whether conditional recovery/exposure paths are fixed or replaced. Monetary comparisons are paired finite differences. Identical marginal PD parameters need not imply identical finite-sample default frequencies after a factor-structure change. No hidden PD uplift, covariance repair, collateral recovery, additional netting or risk-neutral conversion is performed.

Value

CalcResult containing a sealed prepared portfolio, joint default/monetary loss paths with weighted risk distribution, grade and first-default paths with optional marked values, or baseline/stress results with paired changes.

See Also

`assign_default_groups`, `allocate_guarantee_coverage`, `credit_loss_distribution`, `incremental_credit_risk`

[1.5in]

`predict_cure_and_redefault`
Ordered Cure and Subsequent Re-Default Projections

Description

Reuse a frozen Aalen–Johansen model to distinguish reversible state occupancy, sustained cure and first default after cure.

Usage

```
predict_cure_and_redefault(current_episode, model,
    action_conditions,
  horizon_grid, state_recoveries, target, currency,
  extrapolation = "error", context = NULL)
```

Arguments

`current_episode`

Named list: episode_id, state, model_time, known_at and logical ever_default.

`model`

Frozen multistate model with a collections_contract declaring cure_states, default_states, supported_actions, supported_initial_states and action_interpretation equal to observational_conditioning.

`action_conditions`

Named list with action_id and assumption_reference.

`horizon_grid`

Increasing nonnegative elapsed times from model_time, in the model target's units.

`state_recoveries`

Horizon-by-state matrix of conditional period cashflow expectations, in model state order. These must not be cumulative recovery totals.

`target`

Exact frozen physical event target.

`currency`

Explicit cashflow currency.

`extrapolation`

Default error rejects unsupported times and actions. Explicit flat continuation labels unsupported reference-model projections.

`context`

Optional calculation context.

Details

Cure states must already encode the specified probation rule in training data. The augmented projection remembers cure and then marks the first subsequent default permanently. Historical default is never erased. Initial default occupancy does not count as re-default. Model training must be known by the episode cutoff. Action conditioning is observational and does not identify a causal treatment effect. Unsupported flat projections do not change model parameters.

Value

A CalcResult with state occupancy, sustained-cure probability, ordered cure-then-redefault probability, expected undiscounted recovery and support diagnostics.

[1.5in]

rank_credit_reason_codes
Consolidate Stored Model and Policy Reasons

Description

Deterministic same-scale contribution grouping with separately reproduced sealed policy evidence.

Usage

```
rank_credit_reason_codes(contributions, reference_basis,
    reason_dictionary,
  policy_decision = NULL, max_reasons = NULL, context = NULL)
```

Arguments

`contributions`

Records with unique feature_id and finite scalar contribution.

`reference_basis`

attribution_id, model_id, model_version, reference_id, explained_quantity, output_scale (probability, log_odds, score, currency, linear_predictor), baseline, prediction and adverse_direction HIGHER or LOWER. Contributions must reconstruct prediction minus baseline.

`reason_dictionary`

dictionary_id, version, model_reasons and policy_reasons. Every entry has distinct reason_code, description and nonnegative integer priority. Model entries name nonoverlapping known feature_ids. Policy entries bind rule_ids.

`policy_decision`

Optional sealed DecisionArtifact. Matched rules are reproduced, not rerun.

`max_reasons`

Optional positive integer model reason limit; omitted reasons remain in the output.

`context`

Optional calculation context.

Details

Signed transformed-feature contributions are summed within a reason before testing adverse direction. Non-adverse groups are suppressed with backing evidence retained. Model reasons rank by descending adverse net magnitude, then priority and code. Policy reasons rank separately by priority and code, retaining matched rule traces and selected status. Their units are never added to model effects. Unmapped contributions and absolute contribution coverage are explicit. Content hashes detect alteration but do not authenticate evidence. Reasons are descriptive, not causal effects or automatically legally sufficient adverse-action notices.

Value

CalcResult with ranked model reasons, separate policy reasons, suppressed/omitted groups, unmapped contributions, coverage and provenance hashes.

[1.5in]

`rating_system`
Dated local ratings, immutable model versions and historical replay

---

Description

Build typed rating subjects, route profiles, assess evidence, select explicit historical model versions, preserve risk quantities and trace local lifecycle events. Batch inference and replay use frozen declarative recipes.

Usage

```
build_rating_subjects(tables, links, as_of, known_at)
route_rating_profile(snapshot, rules, as_of, known_at)
assess_rating_applicability(snapshot, contract, as_of, known_at)
register_rating_model_version(registry, bundle, registration)
select_effective_model(registry, assignments, profile_id, purpose,
    as_of, known_at)
compose_risk_assessment(assessment_id, subject, components, binding,
    as_of,
                        known_at, calculated_at, effects = NULL)
check_assessment_freshness(assessment, policy, events, as_of,
    known_at)
transition_assessment_state(assessment, event, transitions)
rate_subject_batch(requests, registry, batch_id, known_at)
apply_assessment_event(state, event, dependencies, known_at)
replay_assessment_history(requests, registry, expected, known_at,
                          replay_id = "replay")
```

Arguments

`tables`

Named tables keyed by customer, obligor, facility, issue, transaction or portfolio. Rows: subject_id, version, attributes object, effective_from, effective_to (NULL allowed), known_at. Subject type comes from table name.

`links`

Rows: link_id, relation, from_type, from_id, to_type, to_id, effective_from, effective_to, known_at.

`as_of`

ISO YYYY-MM-DD economic observation date.

`known_at`

ISO knowledge cutoff, no earlier than as_of where both are used.

`snapshot`

Named list with subject_id, subject_type, as_of, known_at, attributes. Each attribute is a named list containing value (possibly NULL), observed_at, known_at and optional conflict flag.

`rules`

Rows: rule_id, profile_id, profile_version, numeric priority, conditions list and effective/knowledge dates. Conditions have field, operator, and value except for present/absent operators.

`contract`

Named applicability contract: contract_id, version, required_features list, allowed_values object, support_bounds object, maximum_age_days object.

`registry`

Hashed registry returned by register_rating_model_version, or NULL to start empty. A CalcResult wrapper is accepted.

`bundle`

Named list: model (frozen ModelArtifact), dependencies object, configuration object. The entire bundle is content-hashed.

`registration`

Named list with matching model_id, model_version, registered_at and optional data-only metadata.

`assignments`

Rows with assignment_id, profile_id, purpose, model_id, model_version, model_hash, effective_from, effective_to, known_at.

`profile_id`, `purpose`

Explicit profile and calculation purpose strings.

`assessment_id`

Unique string for the supplied assessment.

`subject`

Named list containing subject_id and subject_type.

`components`

Rows: component_id, quantity, value, subject_id, subject_type, variant, definition. EAD/EL require currency; GRADE requires scale_id, scale_version and graded_quantity. A different subject needs relationship_id.

`binding`

Named list of nonempty strings: profile_id, profile_version, model_id, model_version, model_hash, input_hash.

`calculated_at`

ISO calculation date, not before knowledge cutoff.

`effects`

Optional rows: effect_id, kind, order, level, input_component_id, output_component_id, reason.

`assessment`

Hashed composed assessment or its CalcResult wrapper.

`policy`

Freshness policy: maximum_age_days, maximum_information_age_days, trigger_types list and optional effective_to.

`events`

Freshness event rows: event_id, subject_id, type, occurred_at, known_at, optionally supersedes_assessment_id.

`event`

Lifecycle event: event_id, type, occurred_at, known_at, reason, gates object. Projection event: event_id, revision string, occurred_at, known_at, dependencies list of type/id objects, payload, optional corrects_revision.

`transitions`

Rows: from_state, to_state, event_type, required_gates list.

`requests`

Rows with row_id, known_at, model_id, model_version, model_hash, recipe and input; see Details.

`batch_id`, `replay_id`

Explicit deterministic run identifier.

`state`

Hashed event state, or NULL to start empty.

`dependencies`

Rows with assessment_id, dependency_type, dependency_id.

`expected`

Prior successful batch rows or rows with row_id, input_hash, recipe_hash, model_hash, result_hash for exact replay comparison.

Details

All schemas use version 1.0. Dates have calendar-day resolution. Effective intervals are [effective_from,effective_to); NULL end means unbounded. Knowledge cutoff is inclusive. Use named lists for objects, unnamed lists for arrays including singletons, and structure(list(), names=character())

for empty objects. Row tables also accept data.frame. Hashes detect changed content; they do not authenticate an author.

Subject types remain distinct. An overlapping effective version creates an explicit conflict and is excluded from the resolved graph. Relationships are legal_obligor, joint_liability, economic_dependency, default_correlation, data_identity, contains, pd_reference. Unresolved link endpoints are reported. No PD, grade or loss is copied between subject levels.

Routing requires matching snapshot economic date and snapshot knowledge no later than the query. Attribute observation cannot follow its knowledge date. Future observations and facts learned after snapshot cutoff are unavailable. Conditions combine with AND. Operators are eq, ne, lt, le, gt, ge, in, not_in, present, absent. Missing values fail all comparisons except absent; ordered comparisons are finite numeric. Membership values are arrays. Empty conditions explicitly match all. Smallest matching priority wins. Equal best priority yields AMBIGUOUS, no match NO_ROUTE, unique match ROUTED with complete trace. No expression evaluation occurs.

Applicability distinguishes missing, unavailable, stale, conflicting and outside training support facts. These yield REFER. A known disallowed applicability field yields NOT_APPLICABLE; otherwise APPLICABLE. Support bounds are inclusive marginal ranges, ages are nonnegative integer days, and required feature names are distinct. These statuses are neither default labels nor lending decisions.

Registry versions are immutable content-bound entries, sorted by model ID/version. Model training knowledge must precede registration. Identical bundle and registration are idempotent; a differing bundle or registration under the same model ID/version fails. Selection requires one effective known assignment and the exact pinned bundle hash registered by historical cutoff. It reports SELECTED, MISSING_MODEL, AMBIGUOUS_MODEL, NOT_YET_EFFECTIVE or EXPIRED_MODEL. Later registered models do not replace an explicit historical version. Dependency/configuration maps participate in the bundle hash and are supplied by the caller.

Composition retains PD, GRADE, LGD, EAD, EL and SCORE as separate quantities and variants. Every definition includes event, positive horizon, time_unit (days/months/years), probability_measure (PHYSICAL/RISK_NEUTRAL), conditioning. PD lies in [0,1]; EAD is nonnegative. LGD and EL are finite and may reflect net gains. Grades name their scale and graded quantity (PD, EL or SCORE). Definition differences within subject/variant, duplicate quantities, unlinked other subjects and mixed currencies are explicit composition conflicts; no automatic inference or aggregation occurs. Effects have kind support/transfer/override/cashflow and unique integer order. Input/output component references must exist, output level must match, and producer order must precede consumer order. Repeated output writes and cycles fail. Caller must identify the same economic effect consistently to avoid hidden double counting.

Composed assessments start DRAFT with history. Freshness checks age, information age, effective expiry and known new material events, including late corrections of older facts. Equality to an age limit remains current. Status is CURRENT, STALE or SUPERSEDED. Historical queries cannot use assessment lifecycle state from a later knowledge date.

Lifecycle state names are DRAFT, SUBMITTED, REVIEWED, FINAL, SUPERSEDED, with explicit caller-supplied transitions. Exactly one transition must match, all required gates must be TRUE, reasons nonempty, and dates/history ordered. Duplicate events fail. Conflicted assessments cannot finalize. FINAL only permits a declared move to SUPERSEDED, which is terminal. Append-only history includes previous assessment hash, old/new state and full event; earlier input objects remain unchanged. FINAL is local processing status and conveys no institutional approval.

Batch requests pin model identity and bundle hash. A recipe has method, version 1.0 and parameters. The application_risk method accepts input=list(snapshot=...) and parameters target plus optional calibrator/offset. Its snapshot cutoff must not exceed request knowledge. The binary_glm method accepts input X (row matrix or row lists) and feature_order; parameters expected_target plus optional offset. This finite dispatcher executes supported library inference only; no user code, callbacks or executable serialization. Model registration must be available by each individual request cutoff, and request knowledge must be available by the batch cutoff. Each unique string row_id retains success or a controlled error and missing value; sorted row order is input-permutation invariant. Hashes bind input, recipe, model, numeric result, sorted request collection and registry.

Event projection matches explicit dependency type/ID pairs and returns sorted unique affected assessment IDs without recalculating or mutating risk/accounts. Same ID/revision/content is DUPLICATE with unchanged state; differing content under that key fails. Corrections have distinct revisions referencing the latest processed revision and retain original history. Future events are DEFERRED; state containing future processed events is rejected for a historical cutoff.

Replay executes supplied pinned historical requests and compares input/recipe/model/ result hashes. Missing bundles or references and changed hashes yield NOT_REPRODUCIBLE with findings; complete agreement yields REPRODUCED. Numerical hashes are exact, so numerical runtime differences remain visible. No current champion or external source is loaded. Reproducibility is not empirical validation.

Value

CalcResult with named-list value. Objects and row arrays are data only. Registry, assessment, batch and event state have canonical SHA-256 content hashes. Outputs are detached R values; consumers reject mutated hashes.

Content-bound assessment pipelines

The unchanged batch signature additionally accepts recipe method `assessment_pipeline`, version `2.0`. Its identical data-only recipe must be stored in the registered bundle's `configuration$assessment_recipe`. Dependencies are ID-keyed `artifacts`, each with kind, content, canonical content sha256 and known_at; kind `innovations` identifies supplied random inputs. Empirical fitting consumes frozen method outputs; numerical market pricing and explicitly driven simulations execute their native granular methods.

Input contains snapshot, ID-keyed datasets (value/as_of/known_at), assessment_id, assessment_known_at, calculated_at, lifecycle_events and dependency events. Snapshot/data/model/artifact information must be available at assessment_known_at. Dataset economic dates cannot follow the subject snapshot. Calculation lies between assessment knowledge and the historical request cutoff; lifecycle events cannot exceed that request cutoff. Later review history never changes the original input information set.

Recipe parameters explicitly contain routing_rules, applicability_contract, steps, components, effects, transitions, event_dependencies and nullable decision. Exactly one applicable route is required before execution. Steps contain step_id, method_id, arguments, argument_types and output_contract. References have the exact object form list(‘$ref‘=list(source="step", id="pd", path=list("value",0))). Sources input/artifact/model/step resolve dataset envelopes, frozen content, primary model under ID primary, or predecessor outputs. Both languages use zero-based array indices. Ready DAG steps run in sorted ID order; unknown predecessors and cycles fail. No user code or user-selected imports are evaluated.

Argument types explicitly select numeric_matrix, numeric_vector, character_vector or logical_ vector conversion. Other arguments retain their typed list structure. Output contracts require shape (scalar/vector/matrix/object/records), unit and nullable target. Optional fields contain path, unit, target and nullable target_path; the latter binds an actual nested target, including scenario sampling law. Component references must match producing quantity/currency and full target. Physical and risk-neutral measures and PIT/TTC sampling laws are never inferred or interchanged. A PRE_ FITTED empirical step requires exactly one artifact argument referencing a frozen output with the same method_id. Registry, lifecycle and recursive batch operations use dedicated stages, not arbitrary DAG calls.

The Shapley operation accepts a data-only predictor with kind FROZEN_PD, model, feature_order and expected_target. A trusted internal generic-PD call is constructed; model identity and probability scale must match its explanation contract.

Composition is followed by declared lifecycle transitions and revisioned dependency event projection. Repeated identical lifecycle events are idempotent; different content under the same event identity fails. A dependency correction retains prior history. Projection does not silently change risk; recomputation uses a new dated request. Optional decision arguments invoke the frozen policy separately, with PD and target equal to a composed PD component.

Version-2 rows include assessment_hash, event_state_hash, artifact_manifest_hash, innovation_ hash (manifest entries tagged innovations) and nullable decision_hash, in addition to legacy input/recipe/model/result hashes. Expected replay rows must contain these fields. Full calculation, composition and ordered history are rerun; corrupt/missing artifacts produce NOT_ REPRODUCIBLE findings. Successful values retain routing/applicability, ordered step argument/value hashes, final assessment, artifact manifest, event state/trace and separate decision. Numerical hashes are exact; cross-runtime last-bit differences remain visible. Hashes establish content integrity, not signature authenticity, empirical validity or institutional approval.

Examples

```
empty <- structure(list(), names = character())
s <- list(subject_id = "O", subject_type = "obligor", as_of = "2024-01-01",
          known_at = "2024-01-01", attributes = empty)
r <- list(list(rule_id = "R", profile_id = "P", profile_version = "1",
               priority = 1, conditions = list(), effective_from = "2024-01-01",
               effective_to = NULL, known_at = "2023-12-01"))
route_rating_profile(s, r, "2024-01-01", "2024-01-01")$value$status
```

[1.5in]

`reconcile_card_authorizations`
Reconcile Dated Card Authorizations and Settlements

Description

Replaces settled reservations by booked money, preserving event identity and separate knowledge time. No total account exposure is inferred.

Usage

```
reconcile_card_authorizations(authorizations, events, policy, as_of,
  known_at, currency, context = NULL)
```

Arguments

`authorizations`

Rows with authorization_id, account_id, approved_at, known_at, sequence, currency, amount and expires_at.

`events`

Rows with event_id, authorization_id, account_id, occurred_at, known_at, sequence, currency, amount and kind. SETTLEMENT requires Boolean authorization_complete; REVERSAL requires reverses_settlement_id; CANCELLATION has zero amount.

`policy`

Named list with policy_id, version, known_at, expiry_boundary equal to EXCLUSIVE, logical late_settlement_allowed, authorization_conversion_factor in [0,1], and positive minor_unit.

`as_of`

ISO observation date.

`known_at`

ISO knowledge cutoff.

`currency`

Explicit common currency.

`context`

Optional calculation context matching dates and currency.

Details

A partial capture of 80 from a reservation of 200 leaves booked exposure 80 and pending authorization 120. Complete capture releases the unused reservation. Reversal reduces booked exposure without reopening the hold. Expiry is exclusive; late settlement needs explicit policy permission. Cancellation closes a hold without refunding a purchase. Overcapture and reversal above the original remaining settlement are rejected.

Amounts are reconciled in integer minor units, with an explicit maximum of 2^40 units for each amount and account aggregate. Binary representation error tolerance in units is max(1e-9, min(1e-3, 8*2^(-52)*abs(amount/minor_unit))). This is not a business rounding rule. The scenario conversion factor is supplied, not an estimated CCF, and can produce fractional minor-unit scenario exposures. Only supplied reservations and their booked charges are covered; account payments and installment conversions require a separate cash ledger.

Value

A CalcResult with authorization and account results, ordered event ledger, excluded records, explicit policy and input/output fingerprints.

[1.5in]

`retail_behavior`
Historical Retail Payment and Revolving Features

Description

Replay point-in-time payment ledgers and separate balance movement, limit changes and minimum/full repayment behavior.

Usage

```
build_payment_history_features(obligations, payments, coverage,
    account_id,
  as_of, information_cutoff, lookback_days, allocation_policy,
  currency, grace_days, context = NULL)
build_revolving_features(balance_history, statements, coverage,
    account_id,
  as_of, information_cutoff, lookback_days, currency,
    context = NULL)
```

Arguments

`obligations`, `payments`

Single-account complete historical contractual ledger with the fields accepted by compute_days_past_due, including pre-window balances, prepayments and explicit payment reversals.

`coverage`

Dated object: status COMPLETE/PARTIAL/UNKNOWN, known_at, observed_from, complete_through.

`account_id`

Explicit matching single-account identity.

`as_of`, `information_cutoff`

ISO economic and information dates; cutoff cannot exceed as_of.

`lookback_days`

Positive calendar-day observation window, excluding its left boundary.

`allocation_policy`

Explicit fifo or explicit payment allocation.

`currency`

Exact reporting currency; no automatic conversion.

`grace_days`

Nonnegative delay for payment assessment only. Contractual DPD is unchanged.

`context`

Optional compatible dated context.

`balance_history`

Rows: snapshot_id, release_seq, account_id, observed_at, known_at, nonnegative balance and credit_limit, activity_status (ACTIVE/INACTIVE/CLOSED), currency.

`statements`

Rows: statement_id, release_seq, account_id, issued_at, due_at, observed_at, known_at, statement_balance, minimum_due, paid_by_due, currency. Amounts are nonnegative and minimum_due cannot exceed the statement payable balance.

Details

Payment histories replay each day using the earlier of the feature cutoff and that historical date. Later-known reversals do not rewrite earlier days. Supply the complete contractual ledger, not only window-dated records. Grace changes only the assessment date, never the actual due date or DPD. Each obligation due inside the window is assessed at due date plus grace. Unelapsed grace and obligations not contemporaneously known are reported as unassessable. Payment ratios aggregate allocated amounts over assessed amounts; a zero denominator is NULL. CURRENT and ARREARS describe unpaid-ledger states and are not regulatory default classifications. Partial/unknown history leaves complete features NULL while retaining known-history summaries.

Revolving snapshots select latest-known revisions and require one resolved record per observed day. Missing daily balances are not carried forward. Complete overlimit-day totals require all calendar dates and complete coverage. A zero limit gives NULL utilization. Explicit inactive accounts differ from undrawn active accounts. Closed accounts require zero residual balance and limit. Utilization change equals balance change divided by the new limit plus previous utilization times the relative limit cut using the new limit denominator. This separates net balance movement from limit effects, not gross draw transactions. Trend is endpoint utilization change per elapsed calendar day.

Statement paid_by_due must mean allocated payments by contractual due date, not eventual receipts. Observation must extend through due date. Full, minimum without full, below minimum and no-amount-due states remain separate. Statement counts refer to observed supplied evidence rather than asserting unseen history exists.

Value

CalcResult containing daily history, assessments and descriptive summaries with explicit missing coverage. Diagnostics retain cutoff policy, coverage and excluded revisions.

See Also

`compute_days_past_due`, `build_bureau_features`

[1.5in]

retail_cashflows
Booked Cashflow and Retail Liquidity Features

Description

Compute locally classified dated cashflows, own-transfer reconciliation, uncertainty, volatility and observed liquidity buffers.

Usage

```
build_transaction_features(transactions, daily_balances,
    category_mapping,
  coverage, customer_id, own_account_ids, as_of, information_cutoff,
  lookback_days, currency, context = NULL)
```

Arguments

`transactions`

Rows: transaction_id, release_seq, account_id, posted_at, value_at, known_at, signed amount, currency, kind ORIGINAL/REVERSAL, category_code. Reversals additionally identify reverses_ transaction_id. Own transfers identify transfer_id and counterparty_own_account_id.

`daily_balances`

Rows: balance_id, release_seq, account_id, observed_at, known_at, signed closing_balance, currency.

`category_mapping`

Dated mapping_id, version, known_at and categories mapping local codes to canonical categories.

`coverage`

Dated status COMPLETE/PARTIAL/UNKNOWN, known_at, observed_from, complete_through.

`customer_id`, `own_account_ids`

Explicit customer and owned-account identities; ownership is asserted, not inferred.

`as_of`, `information_cutoff`

Economic and knowledge dates; cutoff cannot follow as_of.

`lookback_days`

Positive calendar-day window excluding the left boundary.

`currency`

Exact reporting currency.

`context`

Optional compatible dated context.

Details

Latest knowledge-visible releases are selected before the posting-date window. Value dates remain visible. Reversals must reference earlier visible same-account originals of opposite sign and cannot cumulatively exceed them. They inherit the original classification while retaining their own cash date. Supply pre-window originals needed for lineage. Cross-window reversals remain actual cash movements.

Categories are SALARY, OTHER_INCOME, BUSINESS_REVENUE, CONSUMPTION, DEBT_SERVICE, BORROWING, REFUND, OWN_TRANSFER, OWNER_DRAW, TAX and UNKNOWN. Unmapped codes remain UNKNOWN. Only salary and other income enter classified income; consumption, debt service and tax enter classified expense. These are classified subtotals, not claims about the economic nature of unknown amounts. Borrowing, business sales, refunds and owner draws are not household income.

Own-transfer legs must identify two explicitly owned accounts and opposite same-currency amounts. Single visible legs remain unmatched transfers, not income. Own transfers and owner draws are excluded from external cashflow statistics. Daily external net cashflow includes zero posting days; its standard deviation uses the population denominator and numerical scaling. Incomplete history leaves complete statistics NULL while retaining known-record statistics.

Balance snapshots require one resolved observation per account-day. Negative accounts count separately as overdraft account-days. Household buffer observations require all declared accounts that day. Missing balance days are never carried forward or interpreted as zero overdrafts. Full overdraft totals require complete source coverage and every declared account-day.

Value

CalcResult with category credit/debit/net totals, known and coverage-qualified cashflows, daily volatility, category uncertainty, observed liquidity, classified transaction lineage and unmatched own transfers.

See Also

`build_payment_history_features`, `reconcile_income_evidence`

[1.5in]

`retail_evidence`
Dated Bureau, Thin-File and Income Evidence

Description

Describe local credit files, evidence availability and reconciled income without inferring zero debt from missing sources or summing the same salary across sources.

Usage

```
build_bureau_features(snapshots, tradelines, inquiries,
    provider_mapping,
  customer_id, provider_id, as_of, information_cutoff,
    lookback_days,
  max_age_days, currency, context = NULL)
build_thin_file_features(bureau_features, evidence,
    eligibility_profile,
  customer_id, as_of, information_cutoff, context = NULL)
reconcile_income_evidence(declarations, documentary_evidence,
  transaction_evidence, inclusion_policy, customer_id, as_of,
  information_cutoff, currency, period, basis, context = NULL)
```

Arguments

`snapshots`

Rows: snapshot_id, release_seq, customer_id, provider_id, observed_at, reported_at, known_at, coverage_status (COMPLETE/PARTIAL/NO_HIT), currency. Observation precedes reporting and receipt.

`tradelines`

Rows: snapshot_id, snapshot_release_seq, tradeline_id, opened_at, nullable closed_at, status (OPEN/CLOSED/UNKNOWN), nullable balance, limit and integer max_dpd. Amounts and DPD are nonnegative.

`inquiries`

Rows: inquiry_id, release_seq, customer_id, provider_id, occurred_at, known_at, kind (HARD/SOFT/UNKNOWN).

`provider_mapping`

Dated mapping_id, version, known_at and canonical-to-source field maps snapshots, tradelines, inquiries. Empty maps use identity; nullable fields must be explicit.

`customer_id`, `provider_id`

Explicit local identities.

`as_of`, `information_cutoff`

ISO economic and knowledge dates; cutoff cannot exceed as_of.

`lookback_days`, `max_age_days`

Positive inquiry lookback and nonnegative maximum bureau age, in calendar days.

`currency`, `period`, `basis`

Exact monetary currency, income period and NET/GROSS basis. No automatic conversion.

`context`

Optional compatible dated calculation context.

`bureau_features`

Matching customer/date bureau CalcResult or value.

`evidence`

Rows: evidence_id, release_seq, known_at, customer_id, evidence_type, observed_from, observed_until, coverage_status (COMPLETE/PARTIAL/UNKNOWN). BUREAU is reserved for bureau_features.

`eligibility_profile`

Dated profile_id, version, known_at, minimum_tradelines, minimum_history_days, maximum_evidence_age_days, logical allow_partial_evidence and feature_requirements mapping names to lists of required evidence kinds.

`declarations`, `documentary_evidence`, `transaction_evidence`

Rows: globally unique evidence_id, release_seq, known_at, customer_id, component_id, component_type, observed_from, observed_until, nonnegative amount, basis, period, currency, evidence_reference. Revisions retain evidence identity.

`inclusion_policy`

Dated policy_id, version, known_at, source_priority (each DECLARED/DOCUMENT/TRANSACTION once), included_component_types, expected_component_ids, coverage_status, maximum_age_days, logical allow_stale, nonnegative relative_discrepancy_threshold.

Details

Latest knowledge-visible revisions are selected before economic filters. Bureau tradelines bind both snapshot_id and snapshot_release_seq; late corrections cannot rewrite earlier evidence. Inquiry counts use the interval (as_of-lookback_days, as_of] and supplied local history. Closed balances remain separate from open balances. Missing values and partial coverage prevent complete open totals; zero limits give NULL utilization. Absent and no-hit files never establish zero household debt. Maximum reported DPD describes the supplied tradeline history rather than an invented lookback maximum. No provider score is reproduced.

Thin-file strata describe evidence, not default classes. Each feature requires all named evidence kinds with acceptable age and coverage. Supplied thresholds and partial-evidence policy are explicit. Common direct outcome labels cannot be designated input features. Use lists for JSON-style arrays, including single-element arrays.

Income types are SALARY, BONUS, PENSION, RENT_NET, NET_BUSINESS_INCOME and OTHER_NET_INCOME. Each stable component is selected once from its newest eligible source evidence according to the declared priority. Salary and bonus need distinct identities. Business revenue, borrowing, owner transfers and refunds are not inferred as disposable income. Gross/net, period and currency mismatches are excluded without guessed conversion. Document support is supplied evidence, not independent authenticity verification. Source values and discrepancies from the declaration remain visible; a zero declaration has no relative denominator. Incomplete coverage or missing expected components leave total income NULL while retaining the known subtotal.

Value

CalcResult with bureau file state, coverage, age and features; evidence strata and supported/unsupported features; or reconciled components, selected subtotals, nullable total and discrepancies. Diagnostics retain source revisions, policy or mapping fingerprints and excluded evidence.

See Also

`validate_retail_dataset`, `aggregate_household_obligations`

[1.5in]

`retail_features`
Point-in-Time Retail Records and Household Obligations

Description

Validate explicit local retail schemas, freeze typed dated relationships, and separate unique household obligations, budget allocations and person-level legal liability.

Usage

```
validate_retail_dataset(tables, data_contract, as_of,
  information_cutoff, context = NULL)

build_customer_household_graph(entities, relationships, as_of,
  information_cutoff, context = NULL)

aggregate_household_obligations(obligations, household_graph,
  coverage, currency, period, guarantee_budget_policy,
    context = NULL)
```

Arguments

`tables`

Named list of local data frames or row lists.

`data_contract`

Version1.0 object with contract_id, version, table_specs, prohibited_feature_fields and source_coverage. See Details.

`as_of`, `information_cutoff`

ISO economic and information dates. Information cutoff cannot exceed as_of.

`context`

Optional compatible calculation context. Core known_at and information_cutoff aliases, when supplied, must agree with the cutoff.

`entities`

Rows with unique entity_id, entity_type and known_at. Types are CUSTOMER, HOUSEHOLD, APPLICATION, FACILITY and BUSINESS.

`relationships`

Dated release rows with relationship_id, positive integer release_seq, source_id, target_id, role, effective_from, nullable effective_until, known_at, nullable liability_fraction, nullable budget_share, and evidence_reference.

`obligations`

Snapshots with facility_id, release_seq, observed_at, known_at, source_id, debt_stock, scheduled_debt_service, currency, period and status. Amounts are nonnegative or explicitly NULL; status is ACTIVE, CLOSED or UNKNOWN.

`household_graph`

Content-bound graph CalcResult or its artifact value.

`coverage`

Object with status COMPLETE/PARTIAL/UNKNOWN, named sources and known_at. Complete coverage requires at least one source.

`currency`, `period`

Exact reporting currency and debt-service period. No automatic FX or frequency conversion.

`guarantee_budget_policy`

Explicit exclude or include_declared_share. Guarantees never automatically become current debt service.

Details

Each table specification has table, role, primary_key, fields, foreign_keys, date_order, event_field, known_field, feature_fields and allow_extra_fields. Roles are IDENTITY, FEATURE, OUTCOME or DECISION. Fields declare name, type (string/date/number/integer/boolean) and boolean nullable. Foreign keys declare field, table and target_field. Date-order pairs declare earlier and later fields. Keys and features are explicit arrays; use lists for JSON-style arrays, including single-element arrays. Event and known fields must have date types. Source coverage rows declare source_id, status and known_at. Future coverage assessments appear UNKNOWN. Eligible children must reference cutoff-eligible parents. Repeated applications remain separate when application IDs differ. Violations use zero-based row indices in both languages.

Only FEATURE tables can designate model inputs. User-prohibited fields and common direct outcome names cannot be designated features. This contract does not infer the meaning of arbitrary derived variables or establish their lending suitability. Self-referencing payment foreign keys can preserve reversal identity; economic allocation remains separate.

Relationships run from customers to households (MEMBER_OF), applications (COAPPLICANT_ON), facilities (BORROWER_ON or GUARANTOR_ON), or businesses (OWNER_OF). The latest cutoff-known release ordered by known_at and release_seq is selected before filtering its half-open effective interval. Identity is never inferred from address or name. Duplicate active customer/target/role edges and multiple active economic households per customer are rejected.

Debt fractions belong only to borrower/guarantor roles and lie in [0,1] or are NULL. Internal borrower budget shares cannot sum above one per facility. Legal fractions can each equal one for joint borrowers. Household economic membership is not an automatic legal default unit, guarantee or owner-PD transfer.

Aggregation selects the latest known obligation snapshot per facility and counts each linked facility once per household. Legal liability and internal budget allocation remain distinct. Closed facilities require explicit zero residual amounts. Missing snapshots, amounts or statuses and partial/unknown coverage leave totals NULL even if known subtotals equal zero. Missing budget shares separately prevent complete budget totals. Snapshot age, release and source remain visible. Content bindings prevent silent modification of a supplied graph.

Value

A CalcResult. Validation returns valid, applicability, violations, row counts, eligible row indices and source coverage. Graph construction returns a dated, content-bound typed graph, memberships and liability paths. Aggregation returns household known subtotals and nullable totals, budget allocations, individual legal liabilities and source coverage.

See Also

`compute_days_past_due`, `join_as_of`

Examples

```
entities <- list(
  list(entity_id = "person", entity_type = "CUSTOMER", known_at = "2024-01-01"),
  list(entity_id = "house", entity_type = "HOUSEHOLD", known_at = "2024-01-01"))
relationships <- list(list(relationship_id = "membership", release_seq = 1,
  source_id = "person", target_id = "house", role = "MEMBER_OF",
  effective_from = "2024-01-01", effective_until = NULL,
  known_at = "2024-01-01", liability_fraction = NULL, budget_share = NULL,
  evidence_reference = "verified household membership"))
graph <- build_customer_household_graph(entities, relationships,
  "2024-01-31", "2024-01-31")
graph$value$memberships
```

[1.5in]

retail_obligations

Dated BNPL Purchase Plans and Allocated Obligations

Description

Aggregate known purchase plans, payment/refund allocations, separate disputes and explicit provider coverage.

Usage

```
build_bnpl_obligation_features(plans, installments, events,
  provider_coverage, customer_id, as_of, information_cutoff,
  lookback_days, near_term_days, currency, context = NULL)
```

Arguments

`plans`

Rows: plan_id, release_seq, customer_id, provider_id, purchased_at, known_at, currency, nullable actual closed_at, nullable refinances_plan_id, logical schedule_complete.

`installments`

Rows: installment_id, plan_id, release_seq, known_at, due_at, nonnegative amount, currency.

`events`

Rows: event_id, release_seq, known_at, occurred_at, plan_id, nullable installment_id, kind, nonnegative amount, currency, nullable related_event_id.

`provider_coverage`

Dated status COMPLETE/PARTIAL/UNKNOWN, sources (provider IDs), known_at.

`customer_id`

Explicit matching customer.

`as_of`, `information_cutoff`

Economic and information dates; cutoff cannot follow as_of.

`lookback_days`, `near_term_days`

Positive past observation and future due-cashflow windows in calendar days.

`currency`

Exact reporting currency.

`context`

Optional compatible dated context.

Details

Latest knowledge-visible revisions are used. Financial events PAYMENT, MERCHANT_REFUND, PAYMENT_REVERSAL and REFUND_REVERSAL require explicit same-plan installment allocation. No allocation rule is guessed. Reversals identify earlier original events and cannot cumulatively exceed them. Net paid plus refunded amounts cannot exceed the contractual installment; excess customer cash refunds are separate cashflows. Financial events cannot follow actual plan closure. Closed plans require zero residual known unpaid installments.

DISPUTE_OPEN and DISPUTE_CLOSE have zero financial amount. Closing identifies one earlier opening on the same plan. Disputes do not themselves cancel debt or imply default. Explicit refinancing must reference an earlier visible plan; repeated purchases alone are not refinancing.

Concurrency uses actual half-open purchase/closure intervals visible at the feature cutoff, not assumed contractual maturity. It is a retrospective known history rather than a contemporaneous-knowledge replay. Active contracts and positive-outstanding plans are separate counts. Near-term cashflows use (as_of, as_of+near_term_days]; overdue and due-today amounts are separate. Incomplete provider coverage, incomplete schedules or a plan without installments leave total outstanding NULL. Known subtotals remain available.

Value

CalcResult with plan and provider concurrency, repeat use, explicit refinancing, outstanding installments, near-term due cashflows, separate disputes and source-release diagnostics.

See Also

`build_transaction_features`, `aggregate_household_obligations`

[1.5in]

`retail_pool_diagnostics`
Validation of Frozen Retail Risk Pools

Description

Replay dated pool assignments and separately assess mature PD, closed-workout LGD, observed default EAD, concentration, residual heterogeneity and migration.

Usage

```
retail_pool_diagnostics(fitted_model, pool_assignments,
    assignment_inputs,
  predictions, cohort_outcomes, workouts, links, estimation_profile,
  design, previous_assignments = NULL, uncertainty = "none",
  cluster_resamples = NULL, confidence_level = .95,
    tolerance = 1e-10,
  context = NULL)
```

Arguments

`fitted_model`

Frozen retail pooling model.

`pool_assignments`

Sealed retail pool assignments.

`assignment_inputs`

Original assignment arguments except model and context.

`predictions`

Dated validation rows with observation ID, PD, nullable LGD/EAD, frozen reference exposure, explicit training/calibration exclusion.

`cohort_outcomes`

Dated cohort records accepted by estimate_pool_risk_parameters; negative labels additionally require complete_through.

`workouts`

Dated recovery and exposure records accepted by estimate_pool_risk_parameters.

`links`

Observation, facility, obligor, household, risk-group, diagnostic-group identifiers and known-at date. Optional previous_observation_id.

`estimation_profile`

Predated diagnostic weighting profile.

`design`

Common physical validation target, currency and cutoff.

`previous_assignments`

Optional earlier sealed assignments.

`uncertainty`

None or supplied cluster_bootstrap.

`cluster_resamples`

Whole-cluster draws respecting connected units.

`confidence_level`

Interior percentile level.

`tolerance`

Assignment reconciliation tolerance.

`context`

Compatible optional calculation context.

Details

No pool refit is performed. One forecast per declared default unit is required. Whole horizons mature before PD metrics; missing/open workouts do not invent final LGD. Constant pool PD has no identifiable within-pool rank. Relationships never propagate default labels. Observed diagnostic summaries do not replace frozen forecasts. Independent-unit binomial and episode intervals are omitted; supplied whole-cluster draws provide conditional uncertainty. A residual range or sparse sample does not establish pool homogeneity.

Value

CalcResult containing ValidationResult with separate parameter metrics and applicability, residual diagnostics, concentration, migrations, exclusions and cluster intervals.

References

BCBS CRE36, published 2022-12-08, paragraphs 36.16–17 and 36.24. https://www.bis.org/committees/bcbs/basel-framework/standard/cre/36/inforce/2023-01-01/published/2022-12-08

[1.5in]

`retail_pools`
Frozen Retail Pools and Local Risk Parameters

Description

Fit deterministic constrained adjacent score pools, replay frozen assignments and estimate separately weighted local PD, LGD, EAD and CCF with explicit maturity and default conditioning.

Usage

```
fit_retail_risk_pools(observations, pooling_spec, target, training,
  model_id, model_version, context = NULL)
assign_retail_risk_pool(observations, fitted_model, as_of,
  information_cutoff, expected_target, uncovered_policy,
    context = NULL)
estimate_pool_risk_parameters(pool_assignments, cohort_outcomes,
  workouts, estimation_profile, evaluation_cutoff, calibration_id,
  calibration_version, context = NULL)
```

Arguments

`observations`

Training or scoring row lists/data frames. All rows identify observation_id, index_date, feature_observed_at, feature_known_at, declared strata and score. Training adds unique default_unit_id, role TRAIN, label_window_end, label_known_at, logical defaulted, nullable default_date, lgd_status, nullable lgd/ead and their nullable lgd_known_at/ead_known_at.

`pooling_spec`

Dated spec_id, version, known_at, stratum_fields, score_field, segments, minimum_cases, minimum_defaults, minimum_lgd_count, minimum_ead_count, maximum_bin_mean_spread. See Details.

`target`, `expected_target`

Exact target with event default, positive integer day horizon, PHYSICAL measure, conditioning not_defaulted_at_index, default_unit facility/obligor, default_definition and currency.

`training`

Metadata with role TRAIN, known_at, split_id, data_fingerprint and optional data_origin.

`model_id`, `model_version`

Explicit fitted model identity and version.

`context`

Optional compatible dated and target context.

`fitted_model`

Bound retail pool FitResult or frozen model artifact.

`as_of`, `information_cutoff`

Scoring economic and knowledge dates; cutoff cannot follow as_of.

`uncovered_policy`

Explicit unassigned or error.

`pool_assignments`

Bound assignment CalcResult or its value.

`cohort_outcomes`

Rows: observation_id, unique default_unit_id, index_date, label_window_end, label_known_at, outcome_status MATURE/CENSORED/OPEN_HORIZON, nullable defaulted/default_date, non-negative reference_exposure, reference_known_at.

`workouts`

Rows: episode_id, observation_id, default_date, default_known_at, ead_at_default, episode_status OPEN/CLOSED, valuation_known_at, reference_date, reference_known_at, reference_drawn, reference_limit, recovery_cashflows, discount_curve, fx_contract.

`estimation_profile`

Dated profile_id, version, known_at; pd_weighting default_unit/reference_exposure; lgd_weighting episode/ead; ead_weighting episode/reference_exposure; ccf_weighting episode/undrawn; confidence; minimum_pool_cases; independence_assumption independent_default_units.

`evaluation_cutoff`

ISO outcome and workout knowledge cutoff.

`calibration_id`, `calibration_version`

Explicit local parameter-table identity and version.

Details

Strata explicitly include product, borrower_type, facility_type and delinquency_state. Each segment declares segment_id, exact values for all strata, finite score_min/score_max, interior increasing cutpoints and separate pd/lgd/ead monotonicity directions none/increasing/decreasing. Score and strata must not be direct outcome fields. Spread thresholds separately declare nullable

pd/lgd/ead maximum ranges of original nonempty-bin means; NULL disables a restriction. Zero support minima explicitly permit absent conditional-parameter evidence. Use lists for JSON-style arrays, including single-element and empty arrays.

Training features must be known by index. The whole horizon and label receipt must precede training cutoff. Default dates are in (index, horizon end]. Nondefault rows cannot carry LGD/EAD. Only CLOSED workout LGD enters as a numeric LGD label; OPEN final LGD is NULL. Conditional outcome knowledge dates lie between actual default and training cutoff. Missing numeric outcomes have NULL knowledge dates.

Initial bins are left closed/right open with the final endpoint included. Greedy adjacent merging prioritizes undersupported blocks, choosing admissible neighbors by the sum of squared normalized available parameter-mean differences, then the leftmost index on ties. With no support merge available, it considers adjacent monotonicity violations. Every merge must satisfy enabled original-bin mean-spread limits. The complete merge path and separate PD/LGD/EAD diagnostics are retained. This is a local heuristic, not a proof of global optimality or infeasibility. Spread constraints describe between-bin means, not statistical proof of homogeneity. Any remaining violation leaves the affected whole segment INSUFFICIENT_SUPPORT and unassignable. No global monotone LTV law is assumed.

Assignment never refits the artifact. Missing features, unseen strata, out-of-domain scores and unsupported segments remain uncovered. IDs bind model/version, ruleset, segment and merged bin interval. Artifact integrity, target identity, model availability and feature chronology are checked. A dated delinquency change can legitimately change the pool.

Parameter estimation uses mature whole-horizon unit outcomes with exposure weights frozen at index. Censored, unknown and unassigned outcomes are separate exclusions, not nondefaults. A default unit appears once. Workouts must match cohort default and reference date. discount_ recoveries values explicit cashflows at the supplied workout valuation knowledge date; workout_ lgd and realized_ccf preserve raw economic ratios, open-workout incompleteness and undefined zero denominators. One workout per observation is allowed.

Table status describes mature-cohort support only; separate parameter statuses identify defined and absent/zero-weight conditional samples. Weighted estimates retain numerators, denominators, positive-weight counts, observed ranges and independent-episode weighted-mean standard errors; fewer than two contributing episodes leaves the error NULL. Open workouts, missing default workouts, zero-EAD closed workouts and undefined CCF remain separate. Sparse pools retain descriptive estimates with insufficient-support status.

calibrate_grade_pd without priors or monotonic recalibration supplies unweighted exact binomial intervals; low_default_bound supplies a one-sided independent binomial bound. These refer to unit counts, not exposure-weighted PD. Inference is conditional on chosen pools, without pool-selection uncertainty. Local cohort PD is not automatically TTC or long-run PD. Missingness, censoring, dependence and representativeness require separate assessment; these routines do not establish regulatory approval.

Value

FitResult with bound pooling ModelArtifact; CalcResult with bound RETAIL_POOL_ ASSIGNMENTS; or CalcResult with a bound CalibrationArtifact of type RETAIL_POOL_RISK_ PARAMETERS and pool risk table. The latter is not a score transform for apply_calibrator.

See Also

serialize_fitted_model, calibrate_grade_pd, low_default_bound, discount_recoveries, workout_lgd, realized_ccf

---

[1.5in]

retail_sme
Explicit Small-Business and Household Cash Interactions

Description

Separate business operating cash, matched owner transfers, household debt and typed legal liability evidence.

Usage

```
build_retail_sme_features(business_cashflows, owner_drawings,
  household_obligations, liability_links, coverage, business_id,
  owner_customer_ids, business_debt_facility_ids,
  household_debt_service_window, as_of, information_cutoff,
  lookback_days, currency, context = NULL)
```

Arguments

business_cashflows

Rows: transaction_id, release_seq, business_id, posted_at, known_at, signed amount, category, currency. SALES additionally needs counterparty_id; DEBT_SERVICE needs facility_id.

owner_drawings

Rows: drawing_id, release_seq, known_at, business_transaction_id, household_transaction_id, owner_customer_id, business_posted_at, household_posted_at, business_amount, household_amount, currency.

household_obligations

Full aggregate_household_obligations CalcResult or result envelope, with matching dated context and currency.

liability_links

Rows: liability_id, release_seq, known_at, business_id, customer_id, liability_type, effective_from, nullable effective_until, nullable liability_fraction, nullable maximum_amount, evidence_reference.

coverage

Dated history status COMPLETE/PARTIAL/UNKNOWN, known_at, observed_from, complete_through.

`business_id`, `owner_customer_ids`, `business_debt_facility_ids`

Explicit business, owners and business debt identities.

`household_debt_service_window`

Object with from, through, period; dates must equal the cash window and period must match the household result.

`as_of`, `information_cutoff`

Economic and information dates; cutoff cannot follow as_of.

`lookback_days`

Positive calendar-day window excluding its left boundary.

`currency`

Exact reporting currency.

`context`

Optional compatible dated context.

Details

Business categories are SALES, OPERATING_COST, TAX, CAPEX, DEBT_SERVICE, FINANCING, OWNER_DRAW and OWNER_CONTRIBUTION. Signed operating cash comprises sales, operating costs and tax; adding capex and then debt service yields cash before and after business debt. Financing and owner transfers are excluded. Sales are not household income. Owner draws are business debits, contributions credits. Other categories can include explicitly classified signed reimbursements.

Owner drawing legs must match a supplied business posting and opposite household amount once. Local household evidence is asserted, not independently obtained. Unmatched draws remain visible without becoming operating income. Owner households are joined by explicit IDs and counted once. Missing household totals or shared business/household debt facility IDs suppress combined debt subtraction. Cash after household debt excludes living costs and is not affordability or distributable income. No currency or debt-service period conversion is guessed.

Legal types are LIMITED_OWNERSHIP, PERSONAL_GUARANTEE and UNLIMITED_LIABILITY. Fractions lie in [0,1] or are NULL; caps are nonnegative or NULL. Half-open dated links remain explicit evidence; ownership never automatically becomes a guarantee or copied owner PD. Customer concentration uses squared net-sales shares only when total sales are positive and all customer amounts nonnegative. Calendar-month sales and peak-month share are observations, not a seasonal forecast. Negative monthly net amounts make the peak-share statistic undefined.

Value

CalcResult with business cash sequence, owner transfers, monthly and customer sales, matched household debt, overlap exclusions and typed legal evidence.

See Also

`aggregate_household_obligations`, `build_transaction_features`

[1.5in]

`run_tranche_waterfall`
Contractual tranche cash accounts and payment priorities

Description

Allocates actual pool cash to senior costs, interest, liquidity, reserves and tranche principal with explicit triggers and terminal identification.

Usage

```
run_tranche_waterfall(asset_pool_paths, structure_profile,
    context = NULL)
```

Arguments

`asset_pool_paths`

Bound AssetPoolPaths from simulate_asset_pool.

`structure_profile`

Versioned structure and currency; payment_dates; tranches with identity, unique seniority, notional, legal_maturity and dated coupon_fractions; dated additional fees; reserve and liquidity limits; principal priority and diversion rules; OC/IC thresholds and protected seniority; terminal_policy.

`context`

Optional context consistent with physical scenario currency, conditioning and actual calendar-day horizon.

Details

Pool interest and principal/recovery enter separate accounts. Pool workout and servicing costs and nonoverlapping additional deal fees rank first. Tranche coupon and prior arrears follow seniority. Reserve and repayable liquidity provide explicit support. Liquidity repayment precedes reserve refill and tranche principal; it can use principal even where direct principal-to-interest diversion is prohibited. Remaining interest is diverted to principal or equity as configured. Outstanding interest arrears block all principal distributions.

OC uses performing principal, pending net modeled recovery and actual cash over protected outstanding debt. IC uses interest after fees over protected coupon and arrears. A breach forces sequential principal and may authorize principal-to-interest diversion. Ratios with zero denominators are null and do not trigger.

REQUIRE_COMPLETE requires full pool runoff, releases final reserve, disallows new terminal liquidity and records ultimate residual principal write-down. KEEP_OPEN returns null ultimate loss. Timely-interest and legal-maturity failures persist after subsequent cure. No interest on arrears or post-maturity penalty claim is invented. Initial reserve is separately funded support.

Value

CalcResult containing a hashed TrancheWaterfallPaths artifact with dated tranche cashflows, account conservation residuals, persistent payment-failure indicators and ending principal loss identification.

[1.5in]

`scorecard`
Estimated logistic scorecards, odds scales and contribution explanations

---

Description

Fits a logit scorecard using the native GLM engine with separation, rank and convergence checks, then replays contributions and explicitly interpretable score transformations.

Usage

```
fit_scorecard(design_matrix, outcomes, feature_order, fit_spec,
    target, training,
  model_id, model_version, weights = NULL, max_iter = 100,
  tolerance = 1e-10, context = NULL)
score_scorecard(fitted_model, transformed_features, feature_order,
  expected_target, context = NULL)
scale_score_points(log_odds, base_score, base_odds,
    points_to_double_odds,
  odds_convention, score_direction, context = NULL)
apply_qualitative_adjustments(qualitative_inputs, coefficient_spec,
  neutral_levels, base_value, adjustment_scale, context = NULL)
scorecard_reason_codes(score_contributions, reference_profile,
    reason_policy,
  context = NULL)
```

Arguments

`design_matrix`, `transformed_features`

Finite numeric matrix of already frozen WOE/expert features.

`outcomes`

Observed binary labels, default=1.

`feature_order`

Explicit ordered feature names.

`fit_spec`

neutral_profile numeric values and feature_kinds (woe/expert), one per column.

`target`, `expected_target`

Common model target; prediction requires an exact match.

`training`

Common training metadata, including fingerprint, split ID, knowledge date and optional data origin.

`model_id`, `model_version`

Model identity and version.

`weights`

Optional nonnegative integer frequency weights.

`max_iter`, `tolerance`

Native GLM convergence controls.

`context`

Optional calculation context.

`fitted_model`

Scorecard FitResult or ModelArtifact.

`log_odds`

Default-over-nondefault log-odds.

`base_score`

Score at the stated base odds.

`base_odds`

Strictly positive base odds under odds_convention.

`points_to_double_odds`

Strictly positive absolute score change for doubling odds.

`odds_convention`

default_over_nondefault or nondefault_over_default.

`score_direction`

higher_is_safer or higher_is_riskier.

`qualitative_inputs`

Named observed categorical levels. Missing is not neutral.

`coefficient_spec`

Named feature lists of level coefficients on the declared adjustment scale.

`neutral_levels`

Named reference level for every adjusted feature.

`base_value`

Scalar predictor or rating-notch base.

`adjustment_scale`

log_odds or rating_notches.

`score_contributions`, `reference_profile`

Matching named scalar feature-contribution maps.

`reason_policy`

adverse_direction positive/negative, positive integer max_reasons and named reason_codes for every feature.

Details

Preprocessing and supervised binning must already be frozen within the training fold. Feature-kind declarations do not independently establish absence of leakage. The model contains an intercept and neutral-profile contribution reference. Complete or quasi separation is a controlled error; no proprietary coefficients are supplied.

Points are affine in default log-odds, with slope minus PDO/log(2) for higher-is-safer and plus PDO/log(2) for higher-is-riskier. Base odds may use either direction, which is explicitly converted. Returned inverse parameters reconstruct PD without rounding.

Qualitative effects subtract neutral-level coefficients. Log-odds effects return adjusted PD; integer rating-notch differences do not invent a PD or unknown rating-scale boundaries. Reasons compare model contributions with a declared reference, rank adverse differences, break ties lexically and retain full reconciliation. Explanations do not imply causality.

Value

FitResult for fitting; otherwise CalcResult. Scoring returns PD, log_odds, intercept, feature and reference contributions and differences. Scaling returns points and inverse parameters. Adjustments return contributions, total, adjusted value and optional PD. Reasons return selected reasons, all differences, total difference and unreported difference.

References

Native binomial logit estimation follows `stats::glm.fit`; independent tests recover grouped binomial maximum-likelihood estimates and verify score inversion and contribution arithmetic.

[1.5in]

selection_bias
Selection Mechanisms, Reject Inference and Honest Evidence Scopes

Description

Native dated selection panels, logit propensities, sequential inverse probability weights, hypothetical augmentation, binary bivariate-probit selection likelihood, augmented population means and partial-identification diagnostics. Missing own-contract outcomes never become measured validation labels.

Usage

```
build_selection_dataset(applications, decisions, bookings, outcomes,
  as_of, target_population, target, context = NULL)
fit_selection_propensity(X, selected, feature_order, mechanism,
    folds,
  row_metadata, target, training, model_id, model_version,
    context = NULL)
selection_ipw_weights(observed, propensities, target_weights = NULL,
  stabilization = NULL, trim = NULL, context = NULL)
fuzzy_reject_augmentation(outcomes, observed, baseline_pd,
    application_ids,
  augmentation_weight = 1, context = NULL)
parcel_rejected_applications(bands, accepted_defaults,
    accepted_counts,
  rejected_counts, log_odds_shift = 0, uniforms = NULL,
    context = NULL)
fit_joint_selection_model(X_outcome, X_selection, selected,
    outcomes,
  outcome_features, selection_features, target, training, model_id,
  model_version, rho_grid = NULL, selected_rho = NULL,
    exclusion = NULL,
  context = NULL)
estimate_population_risk_aipw(outcomes, observed,
    outcome_predictions,
  propensities, fold_lineage, target_weights = NULL,
    clusters = NULL,
  context = NULL)
reject_inference_bounds(outcomes, observed, target_weights = NULL,
  strata = NULL, risk_bounds = NULL, context = NULL)
reject_inference_sensitivity(baseline_pd, observed, outcomes,
  log_odds_shifts, target_weights = NULL, threshold = NULL,
    context = NULL)
prepare_reject_evaluation(candidates, evaluation_rows, as_of,
    target,
  outcome_comparability = NULL, context = NULL)
```

Arguments

`applications`

List of unique application records containing application_id, entity_id and submitted_at. Grouping entities should identify independent people or households including repeat applications.

`decisions`

Records with application_id, as_of, known_at, policy_version and decision: offered, rejected, withdrawn or not_decisionable.

`bookings`

Records with application_id, as_of, known_at and logical taken_up/funded. Funding requires uptake and a recorded offer.

`outcomes`

For the panel builder, records with application_id, as_of, known_at and outcome_scope. Own-contract records require exact target, status observed/immature/censored and binary outcome only if observed. External-contract records are retained separately. Elsewhere, aligned binary observed labels and missing unobserved labels: use a list containing NULL or a numeric vector with NA on unobserved rows.

`as_of`

ISO audit cutoff; later-known information is excluded from panel construction and rejected for evaluation.

`target_population`

Nonempty description of the target applicant population.

`target`

Common target object: event, positive horizon, time_unit days/months/years, probability_measure PHYSICAL/RISK_NEUTRAL, conditioning. Propensity event must equal mechanism; joint outcome conditioning must be POTENTIAL_OWN_CONTRACT.

`X`

Finite already-transformed selection-feature matrix; an n-by-zero matrix fits an intercept-only selection rate.

`selected`

Numeric zero/one indicator of the specified selection mechanism, or of outcome observation for the joint likelihood.

`feature_order`

Distinct feature names in exact matrix column order.

`mechanism`

decision, takeup, funding or outcome_observation. Supply only rows eligible for this mechanism; for example, takeup is conditional on an offer.

`folds`

List of records with nonempty unique zero-based train and validation indices, in both languages. Entities are disjoint and all training labels must strictly precede validation decisions. Validation rows cannot repeat.

`row_metadata`

Aligned records with entity_id, as_of, features_known_at and label_known_at. Features must be known at decision time.

`training`

Common artifact training metadata: known_at, split_id, data_fingerprint and optional data_origin. Propensity labels cannot exceed the cutoff. Joint data fingerprint is computed from the actual designs and labels.

`model_id`, `model_version`

Nonempty frozen model identifiers.

`observed`

Aligned numeric zero/one indicator that the actual own-contract target is observed.

`propensities`

For IPW, a vector or rows-by-mechanisms numeric matrix of sequential conditional selection probabilities. For AIPW, aligned OOF probabilities. Positive retained target mass requires positive joint probability.

`target_weights`

Nonnegative finite population masses, default one; total mass must be positive. AIPW warmup rows require zero target weight.

`stabilization`

Optional vector with one marginal rate in (0,1] per mechanism. Its product is a common weight multiplier, not an identified target total.

`trim`

Optional inclusive joint-propensity support c(lower, upper), lower strictly positive. Excluding mass explicitly changes the estimand; probabilities are not silently winsorized.

`baseline_pd`

Aligned frozen baseline probabilities in [0,1]. Retain the originating model separately. No refit or pseudo-label validation occurs.

`application_ids`

Unique nonempty application identifiers in input order.

`augmentation_weight`

Nonnegative scalar hypothetical mass per unobserved applicant, divided into good/bad training weights.

`bands`

Unique frozen band identifiers.

`accepted_defaults`, `accepted_counts`, `rejected_counts`

Aligned integer band counts. Accepted counts must be positive, accepted defaults bounded by accepted counts, reject counts nonnegative. Unsupported bands require explicit treatment before this function.

`log_odds_shift`

Scalar or per-band finite assumed log-odds shift of reject risk relative to observed band risk. Zero/one baseline rates remain at their boundaries; no hidden smoothing prior.

`uniforms`

Optional list of explicit per-band uniforms in [0,1), exactly one per rejected record. Produces sampled hypothetical labels; otherwise fractional masses. No RNG is called.

`X_outcome`, `X_selection`

Aligned finite numeric matrices for latent outcome and selection probit indices. Intercepts are fitted. Shared named columns must contain identical values.

`outcome_features`, `selection_features`

Distinct column names for the respective designs.

`rho_grid`

Optional nonempty unique correlations in [-.95,.95] for fixed-assumption likelihood profiles. NULL requests estimated correlation and requires an exclusion.

`selected_rho`

Explicit selected member of rho_grid; required in fixed-correlation mode. No best scenario is selected as validation.

`exclusion`

For estimated correlation, a list with features (selection-only columns) and substantive justification. Columns must add independent rank beyond the outcome design. Validity remains user asserted.

`outcome_predictions`

Aligned OOF outcome-nuisance probabilities for AIPW.

`fold_lineage`

List with rows under row_metadata contract, folds, and nuisance_prediction_origin equal to OUT_OF_FOLD for both supplied nuisance predictions. Positive target-weight rows require an OOF prediction. Externally supplied prediction provenance remains a caller declaration.

`clusters`

Optional dependence-group IDs; defaults to metadata entity IDs. Repeated entities cannot be split across clusters. At least two positive-mass independent clusters are required.

`strata`

Optional aligned textual stratum names, default one all-applicant stratum.

`risk_bounds`

Optional named list of assumed c(lower,upper) missing-risk bounds for existing strata. Unspecified strata use c(0,1). Can encode an explicitly asserted monotonic-risk restriction.

`log_odds_shifts`

Unique finite sensitivity-grid shifts applied only to missing potential outcomes. Observed outcomes are held fixed.

`threshold`

Optional [0,1] population-risk threshold for crossing brackets between adjacent grid points; no unverified interpolation.

`candidates`

Frozen records with model_id, model_hash, training_known_at and explicit training_entities list.

`evaluation_rows`

Unique application_id records with entity_id, as_of, known_at and label_origin. OBSERVED_OWN_CONTRACT requires exact target, funded=TRUE, mature=TRUE and binary outcome. AUTHORIZED_EXPLORATION additionally requires local authorization_reference. SYNTHETIC_TRUTH requires exact target and synthetic data_origin. EXTERNAL_CONTRACT remains a separate proxy scope. HYPOTHETICAL/PSEUDO/UNOBSERVED rows never validate.

`outcome_comparability`

Optional explicit target and transport_assumption for external-contract proxy evidence. Even when declared, the scope remains assumed external transport, not directly observed own-contract risk.

`context`

Optional common calculation context.

Details

The panel preserves offered-but-not-taken-up, taken-up-but-not-funded, immature funded and censored funded strata. The caller constructs maturity/exit-policy targets before joining; partial default-free histories must not be supplied as mature nondefaults.

Propensity fitting calls the native binary GLM primitive, with independent forward/entity folds. Warmup OOF predictions remain NULL, never in-sample replacements. The final frozen model uses the declared training sample. Deterministic separation fails explicitly.

IPW is observed target mass divided by the product of sequential conditional propensities. Support restriction is reported with retained/excluded mass and Kish effective sample size. Conditional MAR and positivity are assumptions, not conclusions of fitting.

The joint binary likelihood is Phi(-b) for unselected rows, Phi2(a,b;rho) for selected defaults and Phi2(-a,b;-rho) for selected nondefaults. Native deterministic 64-node Plackett Gauss-Legendre

quadrature and analytic scores drive bounded L-BFGS optimization, initialized by genuine separate probit fits. No continuous-outcome Heckman correction is substituted. Correlation/coefficient boundaries, tiny likelihood cells at or below 1e-14, separation and failed convergence are not accepted fits. Estimated correlation requires explicit exclusion assumptions; fixed-correlation covariance conditions on the supplied correlation. Observed-information covariance assumes IID observations. Hessian conditioning and weak identification are reported. Generic predict_pd returns unconditional Phi(X beta) for the exact potential-own-contract target and union feature order: outcome columns then selection-only columns.

AIPW averages m+S/e*(Y-m). Cluster influence totals give an empirical sandwich standard error and untruncated normal 95 percent interval. Adequate nuisance convergence and independent clusters are required. Out-of-domain estimates are reported rather than clipped. Double robustness concerns a population mean under MAR, not individual reject PD or general MNAR identification.

Worst-case binary bounds use observed default mass and unobserved mass. Optional restrictions remain assumptions. Log-odds sensitivity reports observed versus assumed components; correlation sensitivity is available through the joint model's fixed-rho profiles. Pseudo outcomes never become evaluation truth. Evaluation-only SplitPlans exclude training-entity or temporal contamination and expose distinct actual, external-proxy and synthetic-engineering scopes. No lending actions or pseudo-label performance rankings occur.

Value

CalcResult with method-specific panel, weights, scenarios, mean/influence uncertainty, partial-identification bounds, sensitivity surface or evaluation-only SplitPlan. Fit functions return FitResult with a common portable ModelArtifact and likelihood/OOF diagnostics. Hypothetical labels are marked validation-ineligible; all-synthetic evaluation carries synthetic provenance.

References

Ehrhardt et al., Reject Inference Methods in Credit Scoring. https://adimajo.github.io/assets/publications/rejectInference.pdf

Binary endogenous-probit sample selection documentation. https://search.r-project.org/CRAN/refmans/endogenous/html/sampselect.html

Robins, Rotnitzky and Zhao (1994), Estimation of Regression Coefficients When Some Regressors Are Not Always Observed. doi:10.1080/01621459.1994.10476818

[1.5in]

`selection_bias_sensitivity`
Validation of Frozen Selection Candidates

Description

Compare common application populations without fitting models; separate funded factual, synthetic and transported-proxy evidence and audit assumption-conditioned selection arithmetic.

Usage

```
selection_bias_sensitivity(candidates, evaluation_rows,
    selection_results,
  design, reference_model_id, uncertainty = "none",
    cluster_resamples = NULL,
  confidence_level = .95, outcome_comparability = NULL,
  tolerance = 1e-10, context = NULL)
```

Arguments

`candidates`

Frozen model records with model identity/hash, training date/entities and application-aligned dated PD predictions.

`evaluation_rows`

Standard dated validation prediction records, application identifiers, funding, label origins, known-at dates, exact target and complete-through negative-label coverage.

`selection_results`

Records binding exact population, target and ordered applications to IPW, BOUNDS, MNAR_SHIFT or AIPW fixed arithmetic inputs and raw results.

`design`

Frozen holdout/external validation design, physical target and evaluation cutoff.

`reference_model_id`

Paired comparison reference candidate.

`uncertainty`

None or supplied cluster_bootstrap.

`cluster_resamples`

Whole-cluster draws with replacement.

`confidence_level`

Interior confidence level for paired percentile intervals.

`outcome_comparability`

Optional explicit external-proxy transport assumption and target.

`tolerance`

Nonnegative absolute and relative artifact reconciliation tolerance.

`context`

Optional calculation context, consistent with the declared target and cutoff.

Details

Whole horizons must mature even for known early defaults. Negative labels require complete coverage. Pseudo labels never become validation truth. Funding differs from acceptance. Each candidate covers the same population; time/entity leakage is excluded. IPW ESS and trimmed mass remain explicit; zero support prevents full-population identification. AIPW retains mandatory frozen out-of-fold nuisance lineage. No selection model is refitted. Model hashes and source declarations are not externally authenticated. Arithmetic agreement does not approve a policy. Brier and AUC contrasts are descriptive, without an implicit multiple-testing acceptance rule. Bootstrap uncertainty conditions on frozen models.

Value

CalcResult with ValidationResult, scoped metrics, paired differences, exclusions, actual cluster intervals, assumption-labelled transport results and reconciliation violations.

References

Ehrhardt et al., Reject Inference Methods in Credit Scoring, sections 1–2. https://adimajo.github.io/assets/publications/rejectInference.pdf

---

[1.5in]

`serialize_fitted_model`
Exchange Versioned Local Model Artifacts

Description

Writes and reads non-executable language-neutral model metadata and numeric parameters with canonical SHA-256 integrity verification.

Usage

```
serialize_fitted_model(fitted_model, destination,
    schema_version = "1.0")
load_fitted_model(manifest_path, allowed_schema_versions = "1.0")
```

Arguments

`fitted_model`

A model named list, or a `CalcResult` containing it. Required fields are `schema_version`, `model_id`, `model_version`, `method_id`, `feature_order`, `parameters`, `preprocessing`, `target`, `training`, `diagnostics` and `provenance`.

`destination`

Explicit local UTF-8 JSON file path. Existing files are never overwritten.

`schema_version`

Supported envelope version, currently `"1.0"`.

`manifest_path`

Explicit local model JSON path. Network URLs and UNC paths are rejected.

`allowed_schema_versions`

Allowed envelope versions; unsupported versions are always rejected.

Details

Features must have distinct nonempty names in prediction order. Parameters, preprocessing, target, training, diagnostics and provenance are named JSON objects. Target includes event, positive horizon, time unit, probability measure and conditioning. Training includes data fingerprint, split identifier and knowledge date. Provenance includes data and parameter origin. Empty objects use `structure(list(), names = character())`; explicit arrays use unnamed lists. All resources are inline. Numeric output uses seventeen significant decimal digits to preserve binary64 values exactly across the JSON exchange. Integral JSON literals outside the exact binary64 integer range are rejected on loading; large floating-point quantities use decimal or exponent notation. The loader rejects duplicate keys, malformed schemas, non-finite numbers and mismatched hashes. A hash establishes integrity, not authenticated authorship or empirical validation. Method-specific prediction functions remain responsible for validating parameter meaning and method compatibility. No RDS, external resources or executable payloads are evaluated.

Value

A `CalcResult` containing path, hash and schema version on writing, or a validated model named list on loading.

[1.5in]

`simulate_asset_pool`
Dated asset pool cashflows with shared default events

Description

Evolves initially performing contractual claims under supplied physical scenarios, including competing prepayment, dilution, dated recovery and financed reinvestment.

Usage

```
simulate_asset_pool(asset_pool, scenario_contract, obligor_paths,
  asset_paths, shared_innovations, context = NULL)
```

Arguments

`asset_pool`

Bound CreditAssetPool from build_credit_asset_pool.

`scenario_contract`

Versioned scenario IDs, normalized weights, increasing ISO dates, currency, physical measure, conditioning, sampling design, explicit event ordering and principal-only reinvestment convention.

`obligor_paths`

Complete scenario/default-group/date rows with conditional_pd and default_channel.

`asset_paths`

Complete scenario/asset/date rows with prepayment probability/channel, dilution and recovery fractions, recovery delay days, recovery and servicing costs, interest multiplier, overdue_days and disputed flag.

`shared_innovations`

Unique scenario/date/channel rows with uniform in [0,1). Referenced channels must be complete and unused channels are rejected.

`context`

Optional result context matching currency, physical measure, conditioning and actual calendar-day horizon.

Details

Default is absorbing and shared across each declared debtor group. Dilution proportionately reduces the remaining schedule. Default precedes coupon; otherwise coupon precedes prepayment or scheduled principal. Gross recovery and workout cost are booked once on their actual future date. Extra recovery dates add cash-only rows, not hazard draws.

Reinvestment purchases already observed warehouse claims after their purchase-date cashflows, using retained principal only. Dated eligibility, balance and cash constraints apply. Failed purchases are not retried; remaining cash is released at the exclusive revolving endpoint. Intermediate legal transfer-out requires sale-price mechanics and is rejected. Unpaid assets and recoveries remain explicit at the horizon; modeled credit loss is distinct from received recovery cash. Future originations and initial workout states are outside this engine.

Value

CalcResult containing a hashed AssetPoolPaths artifact with cashflows, asset events, shared group events, acquisitions, ending states and source bindings.

[1.5in]

`simulate_contract_counterparties`
Shared Counterparty Defaults and Contract Cashflow Interruptions

---

Description

Simulate first counterparty events from supplied conditional default probabilities and uniforms, then apply dated contract cure and replacement rules.

Usage

```
simulate_contract_counterparties(counterparty_paths, contracts,
  cure_uniforms, currency, context = NULL)
```

Arguments

`counterparty_paths`

Contiguous period records with scenario_id, counterparty_id, period_start, period_end, conditional_pd, uniform in [0,1), and probability_measure (PHYSICAL). Initially performing counterparties share one first event across all their contracts. Uniforms must have the appropriate conditional law and declared dependence.

`contracts`

Unique contract records with contract_id, counterparty_id, currency, default_payment_timing (BEFORE_DUE), cure_probability, integer cure_delay_days, logical replacement_enabled, integer replacement_delay_days, nonnegative replacement_payment_fraction, and cashflows (date, nonnegative amount). Disabled replacement requires zero delay and fraction. Payment dates must belong to the counterparty assessment grid, including its initial date.

`cure_uniforms`

Records with scenario_id, contract_id and uniform in [0,1). Each scenario/contract combination must appear exactly once.

`currency`

Reporting currency shared by all contracts.

`context`

Optional compatible physical-probability calculation context.

Details

An event occurs at the first period end with uniform less than its survival-conditional PD, before a payment due at that endpoint. Previously paid cash remains paid. Payments are interrupted during the declared cure delay. Successful contractual cure restores future scheduled payments but does not invent recovery of missed payments. If cure fails, replacement starts after cure delay plus replacement delay and pays the declared fraction. The replacement is assumed performing over the remaining horizon. An instant cure/replacement can preserve a payment despite the recorded counterparty event. Counterparty defaults affect operating receipts; they must not also be counted as borrower defaults. Borrower debt-service, UTP and arrears tests remain separate.

Value

A CalcResult containing counterparty_events, contract_events and individual cashflows with scheduled/received amounts, changes, payment fractions and state, plus input fingerprints.

See Also

`detect_cashflow_default`, `simulate_operating_cashflows`

[1.5in]

`simulate_recovery_waterfall`
Dated Recovery Allocation on Shared Default Events

Description

Allocates supplied ultimate collateral proceeds and costs to legal claims with shared default identities and explicit discount anchors.

Usage

```
simulate_recovery_waterfall(default_events, claims,
    realization_paths,
  recovery_policy, context = NULL)
```

Arguments

`default_events`

Records with scenario_id, default_event_id, subject_id and default_date; one first event per subject and scenario.

`claims`

Records identifying scenario, event, subject, claim, pool and currency, with ead_at_default, legal_ balance, priority and deficit_cost_share. A claim belongs to one pool.

`realization_paths`

Records identifying scenario, event, pool, currency, realization and collateral, with date, gross_ value, realization_fraction, cost, discount_factor and discount_anchor_date. Each collateral identity is consumed once per scenario.

`recovery_policy`

Versioned policy with currency, LEGAL_CAP or UNCAPPED_PRO_RATA allocation, PROCEEDS_FIRST_EXPLICIT_DEFICIT_SHARES cost rule and ULTIMATE_COMPLETE completion declaration.

`context`

Optional matching calculation context.

Details

Legal caps persist across realization dates. Ascending priorities allocate first, equal ranks pro rata to remaining balance. Uncapped participation requires one rank and explicit positive allocation weights. Proceeds pay costs once; only deficits are funded by creditor shares. Net recovery is discounted to the shared default date. EAD is distinct from legal balance. Raw LGD may be negative or exceed one and is undefined at zero EAD. Completeness and legal entitlements are caller-supplied evidence; incomplete workouts must not be represented as ultimate losses.

Value

CalcResult with claim-level economic loss and LGD, dated allocation ledger, pool conservation totals and input fingerprint.

[1.5in]

`simulate_trade_receivables`
Invoice cashflows with shared debtor, seller and servicer events

---

Description

Separates debtor credit loss, commercial dilution, capped seller recourse and aggregate operational losses on observed static invoice claims.

Usage

```
simulate_trade_receivables(invoice_pool, scenario_contract,
    party_mapping,
  party_default_paths, invoice_scenarios, seller_terms,
    servicer_terms,
  shared_innovations, context = NULL)
```

Arguments

`invoice_pool`

Bound static CreditAssetPool of zero-interest bullet invoices.

`scenario_contract`

Versioned physical scenario IDs, weights, dates, currency, sampling design and conditioning.

`party_mapping`

Unique party_id to default_group_id mapping across every debtor, seller and servicer.

`party_default_paths`

Complete scenario/default-group/date rows with conditional_pd and default_channel.

`invoice_scenarios`

Scenario/invoice actual payment date, dilution date/fraction and debtor recovery date/fraction/cost.

`seller_terms`

Scenario/seller settlement, performing and default-estate capacities, default recovery fraction and credit-only or credit-and-dilution recourse scope.

`servicer_terms`

Scenario/servicer remittance date, commingling recovery fraction/capacity and replacement cost.

`shared_innovations`

Exactly used scenario/date/channel uniforms in [0,1).

`context`

Optional compatible physical target and actual calendar-day horizon.

Details

Actual delayed payment stays exposed until payment, with default before same-day payment. Commercial dilution precedes that event and remains distinct; full dilution creates no workout cost. Seller recourse settles after debtor recovery and aggregate capacity is allocated once across eligible invoices. Seller cash goes directly to investors.

Before-failure collections held by a servicer share one capped commingling claim. Collections on or after failure redirect directly. Replacement cost is charged once. Same legal parties share their default group across roles. All cash dates must be within the explicit horizon. Principal less credit, dilution, commingling and costs reconciles to investor cash. Legal enforceability, future originations and early-settlement discounts are not inferred.

Value

CalcResult with bound TradeReceivablePaths, source-separated dated cash, shared events, invoice/seller/servicer attribution and scenario conservation residuals.

[1.5in]

`simulate_transactions`
Execute Frozen Transaction Cash Pools on Shared Scenarios

Description

Combines explicit operating cash flows, loan payments, assets, counterparty interruptions, first-default assessment and recoveries for disjoint complete obligor cash pools. No random numbers are generated internally.

Usage

```
simulate_transactions(transaction_recipes, scenario_cube,
  entity_innovations, parameter_snapshot, context = NULL)
```

Arguments

`transaction_recipes`

List of records with `transaction_id`, `version`, `subject_id`, `parameter_id`, `calendar`, `parameter_hash` and `cube_hash`. Hashes bind the exact canonical parameter object and cube. Each calendar record has `start_date`, `end_date`, `payment_date` and `scenario_time`.

scenario_cube

Physical ScenarioCube or its CalcResult, with schema_version="1.0", kind="ScenarioCube", axes c("scenario","time","factor"), distinct scenario_ids and factor_ids, positive increasing time_grid, aligned nested values, normalized nonnegative weights, units, origin, frequency and sampling_contract. Frequency is monthly, quarterly or annual and times are consecutive year fractions of that frequency. Calendar endpoints advance the origin by the corresponding number of months, preserving end of month. The sampling law is FORECAST_CONDITIONAL, STATIONARY_GAUSSIAN or DECLARED_CYCLE_MIXTURE, with an explicit conditioning scope.

entity_innovations

Unique records scenario_id, date, channel_id, finite value. Asset shocks and counterparty/default or cure uniforms consume declared channels. Channels may be intentionally shared by different drivers; missing and unused records are rejected.

parameter_snapshot

Versioned object snapshot_id, version, parameters, mapping each parameter ID to a cash-pool specification. Required fields are currency, reserve_id, initial_reserve, reserve_targets, loans, operating_contract, operating_base, operating_bindings, assets, default_policy, materiality, external_evidence, recovery, sampling_design, legal_pooling, arrears_interest and payment_allocation. See Details.

context

Optional standard calculation context. Currency and physical probability measure must agree with the declared inputs.

Details

Each loan specifies terms, unique liquidity priority and deficit_cost_share; optional base_rate_fixings and recovery_priority pass to the granular debt/recovery methods. Operating base records use simulate_operating_cashflows; multiplier bindings identify factor_id, matching unit, and transformation level or exp. Going-concern asset disposals are rejected since this recipe does not implement disposal-driven collateral removal. Assets specify parameters for simulate_asset_values, market_binding, innovation_channel, and recovery delay_days, realization_fraction, cost, annual_discount_rate. Recovery uses effective annual discounting over actual delay days divided by 365, anchored at the shared default date.

The legal pooling declaration must be FULL_OBLIGOR_CROSS_COLLATERALIZED. Arrears interest is NO_ADDITIONAL_ARREARS_INTEREST and payment allocation is INTEREST_BEFORE_PRINCIPAL. Contractually scheduled nonnegative coupons accrue; actual payments reduce unpaid interest first, then principal. Actual exposure is distinct from the scheduled closing balance. Negative operating cash is an operating deficit with liquidity priority before debt. Reserve cash is consumed only once and any remaining reserve joins recovery only once.

Materiality specifies nonnegative absolute_threshold, relative_threshold in [0,1], comparison="gt_both" and continuous_between_grid_dates=TRUE. The own material-arrears clock starts at the first material grid endpoint, advancing between consecutive material endpoints. All payments must be represented by the grid; continuity is a supplied assumption. Complete independent outside-obligation evidence is required on every scenario/date, including

origin, with explicit assessed no-indicator rows. The default policy must have obligor assessment scope. Borrower first default is assessed after due-date payments, so payments at that endpoint reduce exposure before recovery. The economic dual test cannot veto independent unlikely-to-pay or material-past-due signals.

Optional `counterparties` specifies `paths`, `contracts`, `cure_draws` and `baseline_receipts="INCLUDED_ONCE_IN_OPERATING_CFADS"`. Path rows name an innovation channel at their period end; cure rows name an innovation channel and innovation date. Granular counterparty cash changes enter the operating bridge once. Shared counterparties must have identical default probability/innovation paths across recipes. Their default probabilities are never added directly to borrower default probability.

The recovery object supplies `policy` for `simulate_recovery_waterfall` and `reserve_recovery="REMAINING_RESERVE_AT_DEFAULT"`. Transactions, obligors, monetary operating contracts, loan facilities, collateral and reserves cannot be duplicated across pools. Common macro factors and innovation channels can be shared deliberately.

Going-concern ledgers stop at the first detected default, whose actual exposure, assets and reserve are then frozen for recovery. Initial defaults are retained as paths but excluded from initially-performing risk estimates. IID_UNWEIGHTED sampling requires uniform cube weights; weighted/dependent sampling does not acquire an IID binomial confidence interval. The supplied forecast/stationary/cycle-mixture conditioning is preserved, not inferred from an average. This scoped composition is not automatic legal pooling or regulatory approval, and does not infer missing outside-default evidence.

Value

CalcResult containing transaction bundles with frozen recipe and parameter hashes, scenario paths and actual first-default exposure, default assessments, recovery allocations, risk estimates and optional counterparty paths. Top-level cube, parameter-snapshot and innovation hashes bind lineage.

See Also

`build_debt_schedule`, `apply_cash_waterfall`, `simulate_operating_cashflows`, `simulate_asset_values`, `simulate_contract_counterparties`, `detect_cashflow_default`, `simulate_recovery_waterfall`, `estimate_simulation_risk`

[1.5in]

`simulation_cashflows`
Contractual Debt Service and Cash Conserving Waterfalls

Description

Build dated debt obligations and allocate actual cash and restricted reserves by explicit priority, retaining unpaid claims across periods.

Usage

```
build_debt_schedule(loan_terms, calendar, base_rate_fixings = NULL,
  context = NULL)
apply_cash_waterfall(cash_periods, waterfall_policy, currency,
  context = NULL)
```

Arguments

`loan_terms`

List with facility_id, currency, positive principal, rate_type (fixed/floating), rate_convention (nominal_simple/effective_annual), day_count (ACT/365F or ACT/360), amortization (bullet/equal_principal/annuity/custom), maturity_date; fixed rates require annual_rate, floating rates require spread.

`calendar`

Contiguous accrual-period records with start_date, end_date and payment_date. Floating periods also require fixing_date at/before start_date; custom amortization requires principal_due. Final payment equals maturity_date.

`base_rate_fixings`

Floating-rate records with unique fixing_date and base_rate; all must be consumed. None for fixed-rate contracts.

`cash_periods`

Scenario/date records with scenario_id, date, currency, nonnegative cash_available, reserve_target and amounts_due. The last is a named list specifying every claim's new due amount, including explicit zeros.

`waterfall_policy`

List with policy_id, version, nonnegative initial_reserve, reserve_priority, excess_reserve_policy (retain/release_before_claims), arrears_policy (carry_without_interest), and claims. Each claim has claim_id, unique priority and logical reserve_eligible. Reserve top-up has its own priority.

`currency`

Single reporting currency for all waterfall amounts.

`context`

Optional compatible calculation context.

Details

Debt interest accrues on opening principal. Nominal simple growth is one plus rate times actual day-count fraction; effective annual growth is (one plus rate) to that fraction. A fixed-rate annuity discounts each payment using the specified accrual factors, supporting irregular periods. Floating annuity resets require explicit principal amounts rather than future-rate perfect foresight. Negative interest is preserved, with net lender refunds separated from positive debt service. Maturity

repayment never assumes refinancing. There are no fees, capitalization or embedded prepayment options.

Waterfalls first optionally release excess reserves, then execute claims and reserve top-up by increasing priority. Eligible claims draw restricted reserves only after current cash. Unpaid amounts carry exactly once without interest. Remaining free cash is distributed to equity. Opening cash plus opening reserves equals payments plus closing reserves plus equity distributions. Reserve movements are not operating revenue. Liquidity shortfall alone does not establish regulatory default: materiality, duration and UTP require the separate event evaluator.

Value

A CalcResult containing the debt schedule and frozen terms, or waterfall periods with payment source, carried arrears, reserves, equity cash and the cash identity residual. Policy/terms content hashes preserve the declared contract.

See Also

`detect_cashflow_default`

[1.5in]

`simulation_drivers`
Explicit Operating Cashflows and Physical Asset Paths

Description

Translate supplied quantity, price, cost and market scenarios into reconciled operating cashflows and physical asset values.

Usage

```
simulate_operating_cashflows(driver_paths, operating_contract,
    context = NULL)
simulate_asset_values(asset_parameters, market_paths,
    context = NULL)
```

Arguments

`driver_paths`

Records with scenario_id, period_start, period_end, currency, base_quantity, base_unit_price, base_unit_variable_cost, base_fixed_cost, volume_multiplier, price_multiplier, variable_cost_multiplier, fixed_cost_multiplier, working_capital_revenue_ratio, cash_taxes, capex, asset_disposals and other_operating_cash. Periods are contiguous, monetary amounts are period totals. Taxes and other operating cash may be signed.

`operating_contract`

List with contract_id, version, currency, period_unit (day/month/quarter/year), quantity_unit, initial_working_capital and logical include_asset_disposals_in_cfads.

`asset_parameters`

List with asset_id, currency, valuation_date, initial_value, salvage_value, initial_age_years, economic_life_years, depreciation_model (straight_line/geometric_to_salvage), idiosyncratic_volatility, probability_measure (PHYSICAL) and positive year_basis_days. Geometric ageing also requires nonnegative annual_depreciation_rate.

`market_paths`

Records with scenario_id, date, nonnegative market_index and innovation. Dates are after valuation. Indices are relative levels with initial value one. Innovations are supplied standard-normal draws with declared joint dependence; no random draws are generated implicitly.

`context`

Optional compatible calculation context.

Details

Revenue is base quantity times volume multiplier times base price times price multiplier. Variable cost scales with quantity; fixed cost has a separate multiplier. Working capital is the declared fraction of period revenue. CFADS subtracts variable and fixed costs, cash taxes, capex and working-capital change, and adds other operating cash and explicitly included disposals. Reserves, funding and debt service are not operating revenues. No universal sector formula, tax-loss rule or interest tax shield is inferred.

Asset ageing applies to initial value less salvage, linearly over remaining life or geometrically with continuous depreciation. Geometric ageing approaches salvage asymptotically; economic life remains diagnostic. Market level and a lognormal idiosyncratic factor multiply the ageing baseline. Log increments are $-\sigma^2 dt/2 + \sigma\sqrt{dt}z$. Unit expected idiosyncratic growth requires the declared innovation law, not arbitrary stress draws. Salvage is not a guaranteed price floor. Physical values are distinct from liquidation proceeds, costs and delay.

Value

A CalcResult containing operating periods with the full CFADS bridge, or asset paths with ageing, market and idiosyncratic contributions, plus frozen contract/parameter hashes and input fingerprints.

[1.5in]

`stress_pool_counterparties`
Counterparty netting, collateral and replacement cash stress

Description

Applies shared physical default to distinct existing monetary claims and supplied signed replacement marks with dated recoveries and no duplicate cash claims.

Usage

```
stress_pool_counterparties(baseline_cashflows, exposure_contracts,
  netting_sets, exposure_paths, collateral_paths, settlement_terms,
  scenario_contract, party_mapping, party_default_paths,
  shared_innovations, as_of, context = NULL)
```

Arguments

`baseline_cashflows`

Scenario/cashflow_id/date/amount rows for existing incoming claims.

`exposure_contracts`

Unique exposure/counterparty/netting identities, role/basis and disjoint linked cashflow IDs. Swap replacement marks cannot duplicate baseline cash.

`netting_sets`

Explicit enforceability, unique received-transferable collateral contract and snapshot/default timing. Unenforceable sets are singletons.

`exposure_paths`

Complete scenario/exposure/date signed_replacement_mark grid; zero added marks on cash-linked claims.

`collateral_paths`

Complete scenario/set/date collateral values, haircuts and known_at dates no later than the snapshot.

`settlement_terms`

Scenario/counterparty estate recovery fraction/capacity, recovery/collateral/replacement delays and additional replacement cost.

`scenario_contract`

Versioned physical weighted scenario and risk-date contract.

`party_mapping`, `party_default_paths`

Unique legal party/group mapping and complete conditional default hazard grid.

`shared_innovations`

Exactly used scenario/date/channel uniforms in [0,1).

`as_of`

Origin ISO date.

`context`

Optional compatible physical calendar-day horizon context.

Details

Default occurs after the known collateral snapshot and before same-day cash. Each unpaid linked cashflow is removed once. Signed replacement marks generate one replacement outlay/inflow, then net positive closeout receives collateral and residual estate recovery; negative closeout is paid once. Estate capacity and additional replacement cost apply once per counterparty across sets. Settlement dates outside the horizon are rejected.

Baseline less adjusted cash reconciles to credit loss plus additional cost. Role loss attribution is proportional to positive gross claims, an allocation convention after netting. Posted collateral is outside this received-collateral method. Legal enforceability and derivative prices are externally supplied, not estimated.

Value

CalcResult with bound PoolCounterpartyStress containing adjusted cashflows, removed monetary IDs, source contracts, shared events, loss attribution and conservation residuals.

[1.5in]

`structural`
Asset-based valuation, terminal default, first passage and empirical calibration

---

Description

Granular structural methods with explicit probability measure and default-event contracts. All analytic valuation inputs are scalars; rates are annual and times are years.

Usage

```
merton_equity_value(asset_value, debt_face, asset_volatility,
    risk_free_rate,
  maturity, payout = 0, currency, context = NULL)
infer_merton_assets(equity_value, equity_volatility, debt_face,
    risk_free_rate,
  maturity, asset_bounds, volatility_bounds, currency, payout = 0,
  tolerance = 1e-9, max_iterations = 1000, context = NULL)
distance_to_default(asset_value, default_barrier, asset_volatility,
  physical_drift, horizon, payout = 0, context = NULL)
merton_default_probability(asset_value, debt_face, asset_volatility,
  drift_or_rate, horizon, measure, payout = 0, context = NULL)
first_passage_pd(asset_value, barrier_spec, volatility, drift,
    horizon,
  measure, payout = 0, context = NULL)
simulate_structural_paths(asset_dynamics, barrier_schedule,
    time_grid,
  innovations, measure, jump_counts = NULL, jump_innovations = NULL,
  bridge_uniforms = NULL, context = NULL)
calibrate_distance_pd_mapping(distances, outcomes, metadata,
```

```
  weights = NULL,
evaluation_distances = NULL, bootstrap_indices = NULL,
confidence_level = .95, out_of_bounds = "error", context = NULL)
```

Arguments

`asset_value`

Strictly positive initial asset value.

`debt_face`, `default_barrier`

Nonnegative terminal debt or threshold in the same currency as assets.

`asset_volatility`, `volatility`

Nonnegative annualized asset volatility.

`risk_free_rate`

Annual continuously compounded risk-free rate.

`maturity`, `horizon`

Nonnegative years; inversion requires positive maturity.

`payout`

Annual continuous payout yield, subtracted from drift.

`currency`

Explicit common currency for monetary valuation inputs.

`context`

Optional consistent calculation context.

`equity_value`, `equity_volatility`

Strictly positive observed equity price and annualized volatility.

`asset_bounds`, `volatility_bounds`

Strictly positive lower and upper inversion bounds.

`tolerance`

Maximum permitted scaled residual for each inversion equation.

`max_iterations`

Positive integer optimizer iteration budget.

`physical_drift`

Physical expected annual asset return before payout.

`drift_or_rate`, `drift`

Physical expected return or risk-free rate according to measure.

`measure`

PHYSICAL or RISK_NEUTRAL, explicitly required.

`barrier_spec`

Named list kind constant or exponential, initial_level, and growth_rate for exponential.

`asset_dynamics`

List asset_value, volatility, drift and optional payout, jump_intensity, jump_log_mean, jump_log_sd.

`barrier_schedule`

Nonnegative barrier at each grid time.

`time_grid`

Strictly increasing years starting at zero.

`innovations`

Paths by intervals matrix of supplied independent standard normals.

`jump_counts`

Same-shaped matrix of independent Poisson(intensity times interval) counts.

`jump_innovations`

Independent standard normals for aggregate conditional log jumps.

`bridge_uniforms`

Optional same-shaped independent uniforms in (0,1), for pure diffusion and positive barriers only.

`distances`

Finite observed distance-to-default signals.

`outcomes`

Mature observed binary default outcomes.

`metadata`

Common calibration target, sample window, split, data origin, population and outcome policy; requires PHYSICAL measure.

`weights`

Optional nonnegative integer frequency weights for fitting.

`evaluation_distances`

Optional evaluation signals; defaults to fitting signals.

`bootstrap_indices`

Optional matrix of complete row resamples, zero-based in both languages. Requires at least two rows and unit weights.

`confidence_level`

Pointwise percentile bootstrap quantile level in (0,1).

`out_of_bounds`

error or clip for fitted mapping support.

Details

Equity is valued as a terminal European residual claim. Inversion jointly matches price and equity volatility using native bounded optimization and refinement; failed equation checks raise NON-CONVERGENCE. It does not estimate physical drift. Distance-to-default returns NULL standardized values and explicit status for deterministic cases. Terminal default includes equality at the terminal debt. First passage includes initial breaches and subsequent barrier crossings and supports constant or exponential boundaries.

Simulation uses exact log steps with compensated compound-Poisson lognormal jumps. Supplied draws determine every path. Brownian bridges detect continuous diffusion crossing against log-linearly interpolated positive boundaries. Bridges with jumps are rejected: grid monitoring may miss jump-path crossings. Returned default times are interval upper bounds. Asset paths remain hypothetical unabsorbed paths after default. Monte Carlo standard errors assume independent paths and exclude model and discretization error.

Empirical distance calibration fits a decreasing isotonic mapping using the common calibration artifact. Optional bootstrap refits use zero-based row indices and type-7 percentile quantiles, clipping each resample to its own support. These pointwise empirical summaries do not guarantee coverage for data-adaptive isotonic blocks. Independent time-separated validation remains necessary.

Value

CalcResult. Equity value has equity_value, delta, d1, d2; inferred parameters have asset_value and asset_volatility; distance results have standardized and log distances and boundary_status. PD functions return a scalar probability. Simulation returns asset_paths, default_time_upper_bounds, survival_curve and survival_standard_error. Calibration returns calibration_artifact, evaluation_distances, pd and optional lower/upper bands.

References

Merton (1974), On the Pricing of Corporate Debt, equations 3b and 10–12, doi:10.1111/j.1540-6261.1974.tb03058.x. Black and Cox (1976), Valuing Corporate Securities, section III, equation 7, doi:10.1111/j.15406261.1976.tb01891.x.

[1.5in]

`structured_credit_pool_foundations`
Dated Structured Pools, Static Tranche Loss and Borrowing Bases

Description

Preserves contractual asset identity, dated ownership and eligibility; computes static attachment/detachment losses and explicitly capped invoice borrowing bases. No legal eligibility or operational credit action is inferred.

Usage

```
build_credit_asset_pool(asset_contracts, pool_assignments,
    eligibility_profile,
  pool_id, as_of, information_cutoff, context = NULL)
tranche_loss_from_pool(pool_loss, attachment, detachment,
    pool_notional,
  loss_unit = "fraction", context = NULL)
receivables_borrowing_base(invoices, borrowing_profile,
  as_of, information_cutoff, context = NULL)
```

Arguments

`asset_contracts`

One snapshot per asset_id, with asset_class, nullable invoice_id, obligor_id, default_group_id, seller_id, servicer_id, currency, origination_date, observed_at, known_at, balance_date, principal_balance, schedule, overdue_days, disputed, dilution_reserve, recourse and collateral_allocations. Schedule rows have date, principal_due and interest_due; collateral rows have collateral_id and allocation_fraction.

`pool_assignments`

Records assignment_id, pool_id, asset_id, known_at, effective_from, nullable effective_until, share, purchase_price and purchase_priority. Intervals are half-open; simultaneous shares across all supplied pools cannot exceed one for any asset.

`eligibility_profile`

Versioned profile_id, version, known_at, currency, product_kind, pool_mode, nullable revolving_until, allowed_asset_classes, maximum_dpd, exclude_disputed and eligible_recourse.

`pool_id`

Target pool identity.

`as_of`

ISO pool or invoice snapshot date. Current balances must match it.

`information_cutoff`

ISO knowledge cutoff no later than as_of. Profile must already be known; later-known assets/assignments are excluded.

`pool_loss`

Nonnegative scalar or vector of pool losses in the declared unit.

`attachment`, `detachment`

Fractions satisfying zero <= attachment < detachment <= one.

`pool_notional`

Positive reference pool notional.

`loss_unit`

fraction or amount. Amount is normalized by pool notional.

`invoices`

Records invoice_id, debtor_id, debtor_group_id, seller_id, servicer_id, currency, issue_date, due_date, balance_date, known_at, outstanding, disputed, dilution_reserve and recourse.

`borrowing_profile`

Versioned common profile plus maximum_overdue_days, exclude_disputed, allowed_recourse, advance_rates (NONE/SELLER), default_group_cap, group_caps, additional_reserve, reserve_scope=ADDITIONAL_NOT_IN_INVOICE_DILUTION and coverage_status (COMPLETE, PARTIAL or UNKNOWN).

`context`

Optional currency/physical-measure-compatible calculation context.

Details

Supported product kinds and claim classes are ABS (CONSUMER_LOAN, AUTO_LOAN, LEASE), RMBS (RESIDENTIAL_MORTGAGE), CMBS (COMMERCIAL_MORTGAGE), CLO (CORPORATE_LOAN), MORTGAGE_COVER (RESIDENTIAL_MORTGAGE, COMMERCIAL_MORTGAGE), PUBLIC_SECTOR_COVER (PUBLIC_SECTOR_CLAIM) and TRADE_RECEIVABLES (INVOICE). The profile may narrow compatible classes. Incompatible routes are rejected. Recourse is NONE or SELLER. These classifications do not certify legal product eligibility or establish borrower/default probabilities.

Pool mode STATIC disallows a reinvestment end. REVOLVING needs an exclusive future end. Only current eligible assignments enter reference principal. Eligible future assignments strictly before reinvestment end form separate warehouse inventory, not current collateral. This inventory consists of pre-existing observed assets, not future originations silently introduced into a historical pool.

Origination <= observation <= knowledge and balance_date <= observation are required. Current pool balances must be dated exactly as_of. Stale balances are not rolled forward by pretending scheduled payments were made. Duplicate assets and invoices are rejected. An obligor cannot

be split among default groups. Principal schedules are increasing, nonnegative and reconcile fully to observed balance; balloons/bullets are explicit. There is no invented refinancing or residual. Collateral allocation fractions summed across visible assets cannot exceed one; this is an allocated-collateral contract, not implicit multiple whole claims with unknown seniority. Assignment share checks use every supplied pool and half-open interval boundaries. Exclusions remain explicit in the artifact.

Static tranche loss fraction is min(max(L-A,0),D-A)/(D-A). Tranche notional equals pool notional times D-A, so L=.15, A=.1, D=.2, pool100 yields tranche notional10, loss5 and tranche fraction.5. Excess pool losses from costs can exceed pool notional, but tranche loss is capped at contractual tranche notional. Invalid or nonrepresentable normalizations fail. Static capital loss is not a timed-payment default event, PD or replacement for a dynamic waterfall.

Borrowing-base eligibility excludes late-known/stale currency, aging, disputed and disallowed-recourse invoices according to profile. Invoice aging is the nonnegative date difference after due date. Eligible outstanding first subtracts invoice dilution reserves. Concentration caps use a fixed denominator equal to total eligible net faces before caps, allocated proportionally within debtor group. They do not solve a retained-pool endogenous concentration constraint. One debtor cannot evade a cap by using several group labels.

Apply recourse-specific advance rates after caps, then subtract the separate nonoverlapping reserve and floor at zero. Net groupG invoices80+100 and groupH100 under a50 percent cap retain140+100; at80 percent advance and reserve5 the base is187. Seller recourse is not risk-free. PARTIAL/UNKNOWN coverage reports a known subtotal with null total borrowing base. No actual account limit is changed.

Value

CalcResult with bound CreditAssetPool or ReceivablesBorrowingBase, or a static tranche loss table. Artifacts retain identity, dates, exclusions, input and content bindings; static output reports pool versus tranche normalization.

References

Securitisations: tranching concentrates uncertainty illustrates attachment/detachment and nonlinear concentration of risk-parameter uncertainty; reviewed 2026-10-13. No regulatory capital algorithm is claimed.

[1.5in]

`survival_models`
Frozen survival, competing-risk and multistate models

Description

Native estimation and data-only event-curve prediction under explicit entry, censoring, landmark, feature and target contracts.

Usage

```
estimate_survival_curve(entry_time, exit_time, event_indicator,
  weight_policy, weights = NULL, confidence_level = .95, target,
  training, model_id, model_version, context = NULL)
fit_cox_ph(start_stop_panel, X, feature_order, ties, strata = NULL,
  covariance = "model_based", target, training, model_id,
    model_version,
  max_iter = 200, tolerance = 1e-9, context = NULL)
fit_aft_model(durations, events, X, feature_order, distribution,
    target,
  training, model_id, model_version, max_iter = 500,
  tolerance = 1e-9, context = NULL)
fit_discrete_hazard(person_period_panel, X, feature_order,
    period_grid,
  link, partial_period_policy, target, training, model_id,
    model_version,
  max_iter = 200, tolerance = 1e-9, context = NULL)
fit_cause_specific_hazards(start_stop_panel, X, event_types,
  feature_order, causes, ties, strata = NULL, target, training,
    model_id,
  model_version, covariance = "model_based", max_iter = 200,
  tolerance = 1e-9, context = NULL)
fit_fine_gray(event_times, event_types, X, cause_of_interest,
  feature_order, censoring_spec, target, training, model_id,
  model_version, max_iter = 200, tolerance = 1e-9, context = NULL)
predict_event_curves(fitted_model, covariate_paths, horizon_grid,
  feature_order, target, initial_state = NULL, event_states = NULL,
  start_time = 0, extrapolation = "error", quantile_levels = NULL,
  context = NULL)
estimate_multistate_transitions(event_history, states,
  allowed_transitions, initial_state, target, training, model_id,
  model_version, context = NULL)
```

Arguments

`entry_time`, `exit_time`

Entry and exit vectors with zero or positive entry strictly less than exit. NULL entry means zero.

`event_indicator`, `events`

Numeric observed zero/one events; missing outcomes are not non-events.

`weight_policy`

frequency: weights represent independently replicated observations.

`weights`

Nonnegative integer frequency weights with positive total.

`confidence_level`

Interior probability for Greenwood log-log KM intervals.

`target`

Full model target: event, positive horizon, time_unit (days/months/years), probability_measure=PHYSICAL, and conditioning. Units are supplied consistently, not converted automatically.

`training`

data_fingerprint, split_id, ISO known_at, optional data_origin.

`model_id`, `model_version`

Nonempty immutable model identifiers.

`context`

Optional calculation context.

`start_stop_panel`

List of entity_id/start/stop/event records; covariate_time is required with features and cannot exceed start. Subject intervals are contiguous and cannot continue after a terminal event.

`X`

Finite numeric design matrix, including n by zero for no covariates.

`feature_order`

Explicit unique feature names in matrix order. Prediction must match the artifact.

`ties`

breslow or efron partial-likelihood ties. Both use explicitly uncentered Breslow baseline increments for frozen prediction.

`strata`

Optional per-row named Cox baseline strata, otherwise all.

`covariance`

model_based or cluster_robust, with subjects as clusters. PH score diagnostics retain an independent-subject assumption.

`max_iter`, `tolerance`

Positive integer iteration bound and positive likelihood convergence tolerance below one.

`durations`

Positive right-censored or event times for AFT.

`distribution`

weibull, lognormal or loglogistic, each fitted by its full censored likelihood.

`person_period_panel`

Counting records plus zero-based period index and interval_end_type scheduled or censor. A positive binary outcome requires a prespecified scheduled window.

`period_grid`

Increasing fitted period boundaries. Every observed interval must lie within its declared period.

`link`

logit or cloglog.

`partial_period_policy`

exclude_censored drops incomplete censored periods, requires boundary entry and rejects partial positive-event windows. cloglog_exposure integrates prespecified exposure using a log-duration offset and requires cloglog.

`event_times`

Positive event/censor times from time zero for Fine–Gray.

`event_types`

List of cause strings or NULL censoring values.

`causes`

Complete distinct competing terminal causes for joint cause-specific estimation.

`cause_of_interest`

Observed selected cause for Fine–Gray.

`censoring_spec`

estimator=kaplan_meier, independence_assumption=independent_within_censoring_group, optional per-subject groups. Reverse-KM censoring is fitted separately per group.

`fitted_model`

Frozen survival ModelArtifact or FitResult.

`covariate_paths`

Empty list for KM/Aalen–Johansen. Regression records have path_id/start/stop/covariates/stratum/covariate_time. Paths must cover the requested horizon contiguously, using the absolute model clock. AFT requires a fixed profile.

`horizon_grid`

Nonnegative increasing elapsed horizons after start_time.

`initial_state`

Known starting state, defaulting to the fitted multistate initial state.

`event_states`

Marked states for a separate absorbing first-hit projection. Defaults to the target event name for multistate prediction.

`start_time`

Absolute landmark on the model clock; prediction conditions on eligibility at this landmark.

`extrapolation`

error rejects unsupported times/risk; flat explicitly carries cumulative hazards or AFT survival after observed support; parametric permits AFT tails.

`quantile_levels`

Optional ordered interior AFT event probabilities, requiring parametric extrapolation. Output quantiles are remaining time conditional on landmark survival.

`event_history`

entity_id/start/stop/from_state/to_state records. NULL or unchanged destination denotes no transition; exact transition times, repeated transitions and delayed entry are supported.

`states`

Unique state names defining matrix order.

`allowed_transitions`

List of distinct directed origin/destination pairs. Actual different-state transitions must belong to this graph.

Details

All risk intervals are open at entry/start and closed at exit/stop. Events at a given time precede censoring there. KM uses a product of one minus observed event/risk fractions, with Greenwood log-log intervals. Boundary intervals are explicitly asymptotic and may degenerate, not exact small-sample bounds. Empty observation-risk intervals are stored; prediction rejects crossings by default or reports explicit flat continuation.

Native Cox fits use survival::coxph, including Efron/Breslow ties, counting-process time-varying covariates and strata. No intercept is identified. The baseline is uncentered Breslow in both languages. The PH diagnostic is an efficient score test for identity-time interactions with coefficient nuisance information eliminated by a Schur complement. Event sums of Schoenfeld residuals, global and individual tests are included; unidentified interaction information is reported. Independent reference tests use survival::cox.zph.

AFT uses log(T)=intercept+X beta+scale*epsilon. Epsilon is minimum extreme value for Weibull, standard normal for lognormal and standard logistic for loglogistic. Usual Weibull/loglogistic shape is reciprocal scale, with distribution scale exp(intercept+X beta). Native fitting uses survival::survreg. Covariance coordinates are intercept, beta, log(scale). Event likelihood contributions are density and censored contributions are survival. Prediction uses stable landmark log-survival ratios and optionally their inverse quantiles.

Discrete hazards include period dummy intercepts. Incomplete censoring is never silently coded as a full-period non-event. The cloglog exposure offset is log(observed duration/full period duration)

under constant within-period intensity and noninformative censoring. Grouped event windows must be prespecified; an exact event time is not a valid endogenous exposure endpoint for a Bernoulli likelihood. Logit predictions require complete periods; cloglog allows fractional periods.

Cause-specific estimation censors other terminal causes but integrates all fitted hazards jointly. For a step with total hazard increment dH, survival S is multiplied by exp(-dH), and cause k gains S*(1-exp(-dH))*dH_k/dH. This declared exponential-increment convention preserves total probability and is different from empirical AJ transition fractions. One minus cause-specific survival is not a competing-risk CIF.

Fine–Gray retains competing failures after their observed time T with IPCW G(t-)/G(T-), using group-specific reverse KM. Terminal failures are removed before tied censoring events in that censoring model. The expanded weighted counting-process fit uses Efron ties and an uncentered Breslow baseline. Only fixed baseline covariates and right censoring from time zero are supported; no post-competing-event feature path or delayed-entry IPCW law is invented. Stored covariance is conditional on estimated IPCW; sampling inference requires subject resampling and censor-model refitting. The complement of the selected-cause CIF is not all-event survival, and independently fitted cause CIFs are not assumed jointly coherent. Native expansion and coefficients are tested independently against survival::finegray.

Aalen–Johansen multiplies I+dA, where off-diagonal increments are observed origin-to-destination counts divided by origin risk sets and diagonals are negative row sums. The artifact includes event times, risk sets, transition increments/products and occupancy. Subject intervals must be contiguous and state-consistent; unchanged/censored rows remain risk intervals. Transition probabilities conditional on a landmark/state require a time-inhomogeneous Markov interpretation, or adequate state expansion. Interval-censored state snapshots are not exact transition histories. Missing-risk rows have identity increments with unidentifiable intensity: default prediction rejects positive mass with unsupported outgoing risk, while flat continuation records those assumptions.

Current default-state occupancy may fall after cure. A separate absorbing projection of marked states yields monotone first-hit incidence. For redefault after cure, use a cured initial state, matching absolute landmark/clock and default as the marked state. No causal action effect is inferred.

Prediction returns elapsed horizon_grid, start_time and a list of curves. Curves contain path_id, survival, cause-keyed cumulative_incidence and marginal_probabilities. AJ adds states, horizon-by-state occupancy, transition matrices and first_hit_probability. Optional AFT quantiles add quantile_levels and conditional_event_quantiles. Feature/target mismatches, unknown strata and unsupported default extrapolation fail explicitly. Public data-only artifacts are exchanged and replayed in both languages.

Value

FitResult with ModelArtifact for estimators; CalcResult with frozen conditional event curves for prediction. Parameters and diagnostics preserve risk, censoring, likelihood and prediction conventions.

References

R survival documentation: https://stat.ethz.ch/R-manual/R-devel/library/survival/html/survfit.formula.html, https://stat.ethz.ch/R-manual/R-devel/library/survival/html/survreg.html, https://stat.ethz.ch/R-manual/R-devel/library/survival/html/cox.zph.html, https://stat.ethz.ch/R-manual/R-devel/library/

survival/html/finegray.html. Competing events and multistate product integrals: https://stat.ethz.ch/R-manual/R-devel/library/survival/doc/compete.pdf. Python Cox backend: https://www.statsmodels.org/stable/generated/statsmodels.duration.hazard_regression.PHReg.html.

[1.5in]

synthetic
Offline Synthetic Credit Worlds with Shared Actual Innovations

---

Description

Native R generators and audits for linked macro, wholesale, facility, retail, asset-pool and system-event worlds. Hidden DGP truth is segregated from observations, frozen predictions, decisions and observed outcomes.

Usage

```
generate_macro_reference_world(specification, innovations,
  output_directory, chunk_rows = 1000L, memory_budget_mb = 256L)
generate_obligor_panel(macro_world, specification, innovations,
  output_directory, chunk_rows = 1000L, memory_budget_mb = 256L)
generate_facility_workouts(obligor_world, specification,
    innovations,
  output_directory, chunk_rows = 1000L, memory_budget_mb = 256L)
generate_retail_reference_world(macro_world, specification,
    innovations,
  output_directory, chunk_rows = 1000L, memory_budget_mb = 256L)
generate_asset_pool_reference_world(worlds, specification,
    innovations,
  output_directory, chunk_rows = 1000L, memory_budget_mb = 256L)
generate_system_event_reference_world(worlds, specification,
    innovations,
  output_directory, chunk_rows = 1000L, memory_budget_mb = 256L)
generate_challenge_cases(case_ids = NULL, context = NULL)
validate_synthetic_world(world, check_hashes = TRUE, context = NULL)
```

Arguments

`specification`

Named list with `tier` (tiny, standard, stress), `dgp_family` (well_specified, misspecified), ISO `start_date` and `observation_cutoff`. Retail additionally requires `selection_mechanism`: MAR, MNAR or missing_overlap. All other DGP coefficients are fixed engineering parameters documented in the world, not estimated defaults.

`innovations`

Directory containing a versioned manifest and actual uint32 little-endian stream files. Each stream declares relative filename, two-dimensional shape and SHA256. Uniforms equal (uint32 + 0.5) /

2^32; paired Box–Muller transforms produce normal innovations. Python and R read the same bytes, not matching seeds.

`output_directory`

Absent or empty caller-selected directory. All output is written there; there is no implicit installed-resource write. Existing nonempty output is rejected. An interrupted directory can be inspected; use a fresh directory for a new run.

`chunk_rows`

Integer from 1 to 50,000. Table processing and output are bounded; scenario paths are streamed individually. Changing this does not change draws.

`memory_budget_mb`

Explicit planning budget, at least 32 MiB. A stream/index estimate is checked before output creation. This is not an operating-system process memory limiter. No silent subsampling occurs when declared capacity is insufficient.

`macro_world`

Generated macro result, its value, or directory. Source tier, start and observation cutoff must agree with the requested output.

`obligor_world`

Generated obligor result, its value, or directory; its observed default events seed linked facility workouts.

`worlds`

List of generated worlds or directories. Pool composition uses actual funded accounts/facilities. System events use existing obligors, accounts, facilities and pools. Source identities and dates are preserved.

`world`

Generated result, result value, or directory to audit.

`check_hashes`

Logical scalar. FALSE skips file digest checks only; semantic, temporal, foreign-key and economic checks remain active.

`case_ids`

NULL for all constructed cases, or unique case identifiers. Identifiers are complete_separation, zero_default, spiegelhalter_half, singular_covariance, cyclic_support, transfer_extremes, censored_target, raw_lgd_bounds, cohort_100 and future_feature.

`context`

Optional standard calculation context for challenge/audit results.

Details

Five directories separate observable_inputs, latent_truth, frozen_predictions, decision_history and observed_outcomes. Tables are gzip JSONL. The macro scenario cube is float64 little endian in path/month/factor order and starts after the final history date. All file paths in manifests are relative. Shared innovations are validated before generation; no RNG, network, Python bridge, implicit fit or outcome-driven reselection is used by these functions.

Wholesale tiny/standard/stress comprise 64/3000/20000 obligors over 3/10/15 forecast years; macro history has 36/240/360 months. Scenario dimensions are 512x24x8, 16384x60x8 and 131072x120x8. Standard includes 750 distinct groups. Retail application capacities are 320/50000/500000 over 12/36/84 origination months; household counts are 160/20000/200000 and persons 240/30000/300000. Funded account capacities are 160/24000/240000 with 24/60/120 account months. Pools have 2/12/100 pools and 80/30000/100000 source-asset capacity, four tranches each. Capacities do not force acceptance, survivor counts, default rates or recoveries.

Group/household modulo-10 partitions provide 60/20/20 entity train/calibration/test splits. Forward time partitions are also 60/20/20; mismatched cells are excluded. Annual financial knowledge lags observation by 90 days. Default reporting lags 14 days; recovery reporting lags seven days. Macro releases have initial/revised lags of 30/90 days. Late bureau corrections cannot enter original application predictions. Immature unresolved labels remain NULL, and unfunded applicants never receive own-loan observed labels. Early closure is a competing nondefault.

Balance sheets, earnings and CFADS reconcile. Observable structural equity and volatility are generated using the European-call equations from latent asset values. Independent noisy external opinions, expiring overrides, scoped support, operating leases/charters/offtake and independent foreign-currency transfer sensitivity are separate tables. Zero/low-default strata are explicit.

Facility conversion, EAD and recovery share borrower-risk dependence. Gross recoveries/costs are dated at 90, 360 and 900 days; incomplete workouts remain open. Raw discounted LGD is not clipped. Constructed LGD -0.2 and 1.2 fixtures are excluded from stochastic cohort frequencies.

Retail includes mortgages, installment/auto loans, cards, overdrafts, BNPL, small-business credit and leasing; net-income observations, co-borrowers, bureau tradelines and transaction transfer/reversal links are explicit. MCAR/MAR/MNAR missingness, MAR/MNAR/no-overlap acceptance, take-up, changing policy and funding are separated. Frozen demonstration coefficients use observable data with declared fallbacks, while latent frailty affects only the hidden DGP and MNAR mechanism. Monthly cash ledgers reconcile opening balance plus draws and accrued interest minus payments and recoveries to closing balance. Floating mortgage resets, balloons, missed payments, active undrawn balances, cure/redefault episodes, competing exits and collection actions are dated. Collection actions are a fixed engineering process, not causal treatment-effect estimates.

Pool servicing copies actual source ledgers; contractual future schedules are a separate no-default/no-prepayment projection. Original assigned principal is not current EAD. Forward additions and final aggregate-notional knowledge dates are explicit. System-event checkpoints follow an independently authored closed-form schedule with duplicates, late corrections, failed batches, guarantee expiry and model switch; replay is not used to compute expected checkpoints.

All tiny worlds and full wholesale standard worlds with actual innovations are installed under extdata/synthetic_tiny and extdata/synthetic_standard. The native extdata/synthetic_innovations.R script can export additional complete tier streams offline: Rscript SCRIPT DIR standard –retail. Its SHA256-counter construction matches the Python authoring script byte for byte and differs intentionally from the bundled fixed PCG64 reference experiment. Large tiers require sufficient disk space and explicit run budgets; no data are downloaded.

System-event worlds include model, policy and calibration version registries and complete source snapshots with audited content hashes. Checkpoints are demonstration fixtures, not fitted models. Servicing includes original lifecycle events. A separately constructed open-workout fixture is excluded from stochastic default counts. Exact snapshot hashes are checked against their own sources; independent floating-point calculations may differ by one representable unit.

Audit checks hashes, counts, columns, local identities/foreign keys, date ordering, hidden-truth leakage, funded-only labels, splits, balance/cash/limit identities, pool uniqueness and tranche totals. Conditional hazard sums and realized events are descriptive; dependent observations are not treated as independent binomial samples. Synthetic engineering evidence cannot establish actual portfolio predictive performance, regulatory compliance or causal identification.

Value

A CalcResult. Generators return a value with directory and manifest. Challenge generation returns cases, constructed flag and independent-oracle origin. Audit returns passed, issues, row counts, realized events and descriptive design-versus-realized diagnostics. No audit result implies real-bank validation.

References

Merton, R. C. (1974), On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. https://dspace.mit.edu/entities/publication/5189ad6f-77e1-410e-8b71-29347b2ceb71. NIST, Exact Binomial Confidence Limits. https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/exacbici.htm.

Examples

```
cases <- generate_challenge_cases(c("zero_default", "raw_lgd_bounds"))
cases$value$cases
root <- system.file("extdata", "syn_t", package = "creditratingsystem")
if (nzchar(root)) validate_synthetic_world(file.path(root, "retail"))$value$passed
```

[1.5in]

`transfer_events`
Explicit Payment Restrictions and Transfer Default Risk

Description

Match dated payment restrictions, map events to payment default, combine default events, and integrate exclusive state losses.

Usage

```
assess_transfer_applicability(payments, event_policy, as_of,
    known_at,
  horizon_end, context = NULL)
transfer_event_default_probability(transfer_event_probability,
  event_to_default_probability, mitigation_contract, event_id,
```

```
    target,
  context = NULL)
combine_default_transfer(pre_transfer_risk, transfer_risk,
    dependence,
  joint_probability = NULL, context = NULL)
transfer_loss_components(combined_risk, conditional_losses,
    currency,
  applied_loss_effects, context = NULL)
```

Arguments

`payments`

Payment rows with IDs, due and knowledge dates, source and settlement currencies, payer/payee and route jurisdictions.

`event_policy`

Versioned named policy with explicit event types, jurisdictions, currencies, validity intervals, knowledge dates and evidenced exemptions.

`as_of`, `known_at`, `horizon_end`

ISO dates for the inclusive payment window and information cutoff. Event validity ends exclusively.

`transfer_event_probability`

Scalar event probability for the declared exposure target.

`event_to_default_probability`

Conditional unmitigated default probability given the event.

`mitigation_contract`

Named list with effect_id, success_probability_given_event_and_unmitigated_failure, conditioning equal to event_and_unmitigated_failure, and assumption_reference.

`event_id`

Explicit event identifier.

`target`

Named exposure_id, positive horizon, time_unit (days/months/years), probability_measure (PHYSICAL/RISK_NEUTRAL), and conditioning.

`pre_transfer_risk`

Named stage PRE_TRANSFER, pd, target and applied_effects.

`transfer_risk`

Transfer-default result with a compatible target.

`dependence`

Explicit or independent, in lowercase.

`joint_probability`

Intersection probability within the Frechet bounds; required for explicit dependence and absent for independence.

`combined_risk`

Combined event-union result with stage TRANSFER_ADJUSTED.

`conditional_losses`

Named monetary losses for neither, default_only, transfer_only and joint. Each is a joint conditional expectation, with discounting and recovery assumptions supplied by the caller.

`currency`

Reporting currency.

`applied_loss_effects`

Unique economic effect IDs; cannot repeat PD-level effects. Empty list explicitly means none.

`context`

Optional calculation context.

Details

Transfer default probability equals event probability times conditional failure times one minus conditional mitigation success. With base PD p, transfer PD q and intersection j, combined PD is p+q-j. Each exclusive state contributes exactly once to loss. Equal currencies do not exempt payment freezes. Missing applicability information is reported explicitly. Applicability screens the route at the payment due date; separate timed payment legs require separate rows. A knowledge cutoff later than as_of creates a retrospective view; historical backtests require the original available knowledge cutoff. No legal inference, automatic grade translation or joint loss averaging is performed. The caller supplies probabilities for identical targets and consistently identifies economic effects.

Value

A CalcResult with applicability findings, a probability decomposition, an exclusive state partition, or expected monetary loss and state contributions.

[1.5in]

`validate_probability`
Validate Probability Values

Description

Checks the probability domain without clipping or coercing text or logical input.

Usage

```
validate_probability(value, boundary_policy = "closed",
    context = NULL)
```

Arguments

`value`

A finite numeric scalar, vector, matrix or array.

`boundary_policy`

`"closed"` admits zero and one; `"open"` excludes both endpoints.

`context`

Optional calculation context.

Value

A `CalcResult` with the validated value and its dimensions preserved. Invalid inputs raise a `creditratingsystem_error` condition with a stable `code`.

Examples

```
validate_probability(c(0, 0.2, 1))$value
```

[1.5in]

`validate_rating_scale`
Validate an Explicit Rating Scale and Assign PD Grades

Description

Checks a complete user-defined probability scale and maps probabilities to its uniquely binding grade. No proprietary or default scale is supplied.

Usage

```
validate_rating_scale(rating_scale, boundary_convention = NULL)
map_pd_to_grade(pd, rating_scale, context = NULL)
```

Arguments

`rating_scale`

A named list with `schema_version = "1.0"`, nonempty `scale_id` and `version`, `boundary_convention`, and an ordered unnamed list `grades`. Each grade is a named list with `grade`, `lower`, `upper`, `representative_pd` and logical `is_default`. A scale validation result is also accepted.

`boundary_convention`

Optional assertion of `"left_closed"` or `"right_closed"`. If supplied it must match the scale.

`pd`

A finite scalar or vector of probabilities in [0,1].

`context`

Optional context named list or context-building result.

Details

Grades run from lower to higher risk, cover [0,1] without gaps or overlaps, have distinct identifiers, and have strictly increasing representative PDs within their own intervals. Left-closed intervals contain their lower boundary; right-closed intervals contain their upper boundary. Zero and one are always covered. A final default singleton [1,1] is permitted and takes precedence at one under either convention. Only the final grade may have the default flag. That flag describes a supplied scale label and does not establish an observed default event. A grade is not itself a point probability.

Value

A `CalcResult` with a validated scale or character grade vector. Mapping diagnostics contain scale identity, convention and binding intervals.

Examples

```
s <- list(schema_version = "1.0", scale_id = "example", version = "1",
  boundary_convention = "left_closed", grades = list(
    list(grade = "A", lower = 0, upper = 0.05,
         representative_pd = 0.01, is_default = FALSE),
    list(grade = "B", lower = 0.05, upper = 1,
         representative_pd = 0.2, is_default = FALSE)))
validate_rating_scale(s)$value$scale_id
map_pd_to_grade(c(0, 0.05, 1), s)$value
```

[1.5in]

`validate_source_lineage`
Audit Source Lineage and Frozen Historical Replay

Description

Checks actual row-level source lineage, join allocation and dated use, and compares actual supplied replay checkpoints with immutable original checkpoints.

Usage

```
validate_source_lineage(nodes, edges, joins, reconciliations, uses,
  contract, context = NULL)
replay_consistency(event_log, original_checkpoints,
    replayed_checkpoints,
  snapshot_manifest, replay_contract, context = NULL)
```

Arguments

`nodes`

Source/transform/feature/target/prediction/decision nodes with hashes, schemas, vintages, revisions, dates and actual weighted records.

`edges`

Parent-child lineage edges; the graph must be acyclic.

`joins`

Equality-join mapping ledgers, declared cardinalities, excluded IDs and conservative left-weight/left-amount allocations.

`reconciliations`

Independently expected record identities, total weights and material amounts.

`uses`

Dated uses with purpose, observation/knowledge cutoffs and exact ancestral source bindings.

`contract`

Versioned required node/use/join IDs, separate amount-field currencies and comparison tolerances.

`event_log`

Immutable event deliveries with IDs, dates, sequence, idempotency keys, payload hashes and append-only revision references.

`original_checkpoints`

Frozen original states, dates, applied event IDs and log/snapshot bindings.

`replayed_checkpoints`

Actual replay outputs with identical contracts and explicit REPLAY mode.

`snapshot_manifest`

Frozen model, calibration, policy and source snapshots with hashes, knowledge and effective intervals.

`replay_contract`

Explicit stable ordering, identical-delivery idempotency, append-revision semantics, required checkpoints and numeric tolerances.

`context`

Optional calculation context.

Details

Lineage audits compute totals from supplied rows. Join business-key multiplicities, all matching pairs, unmatched IDs, weights and amounts are checked. Retained left-row allocation fractions sum to one; duplicated full amounts cannot be hidden by a successful technical join. Predictive paths exclude future observations and outcome/post-decision ancestors.

Replay compares actual outputs rather than executing an alternative event engine. Historical event eligibility follows occurrence and knowledge dates. Identical duplicate deliveries are applied once; conflicting duplicates are errors. Later revisions cannot alter earlier knowledge states. Both original and replay applied sequences are checked, together with frozen model/calibration/policy/source snapshots. New-method recomputation is not historical replay. Explicit lists preserve JSON array distinctions in native R.

Value

A CalcResult with a ValidationResult containing individual violations, totals or state differences, fingerprints and PASS/BLOCKED status for the declared audit contract. Upstream provenance is not authenticated and original decisions are not modified.

[1.5in]

`validate_system_coverage`
Assess Explicit System and Release Evidence

Description

Read-only assessment of five-axis coverage, predeclared validation criteria and exact-candidate installed execution evidence.

Usage

```
validate_system_coverage(cells, function_inventory, journeys,
    evidence,
  scope, context = NULL)
validation_report(results, acceptance_policy, manifests,
    context = NULL)
release_candidate_assessment(candidate, evidence, requirements,
  context = NULL)
```

Arguments

`cells`

Unique explicit five-axis applicability cells with contract, function, journey and evidence links.

`function_inventory`

Function identifiers, implementation statuses and native language coverage.

`journeys`

Journey identifiers and covered function identifiers.

`evidence`

Referenced coverage evidence or exact-candidate installed run evidence.

`scope`

Versioned customer-segment, product, lifecycle, method and granularity levels with explicit required cell IDs.

`results`

Named-by-ID ValidationResult records with origin, reference and model/data/split manifest bindings.

`acceptance_policy`

Predeclared required results and separate LE, GE, EQ or BETWEEN scalar acceptance criteria.

`manifests`

Hashed model, data and split manifests and evaluation start date.

`candidate`

Exact candidate ID, wheel/sdist/native-R SHA-256 hashes and resource hashes.

`requirements`

Required cells, journeys, generators, profiles, resources and positive runtime/memory budgets.

`context`

Optional calculation context.

Details

Coverage does not infer a Cartesian product. Required scope cells cannot be waived by changing an applicability record. Missing negative routing, native R or independent-oracle evidence is a visible gap. Non-applicability requires a versioned justified decision and cannot mean missing implementation.

Report policies must precede evaluation. Missing or non-applicable required results and undefined metrics remain blocking. P-values are not combined; synthetic evidence is separately tagged.

Release gates bind every record to the exact artifact and resource manifest. Mandatory installed runs must be executed, offline and isolated from the source tree, separately for Python wheel, Python source distribution and native R. Missing, failed or non-applicable mandatory evidence cannot pass. Measured runtime and memory must satisfy declared budgets. The assessor evaluates supplied evidence without executing tests or publishing.

Value

A CalcResult with a ValidationResult, individual findings, explicit open issues, fingerprints and a PASS or BLOCKED decision confined to the declared evidence contract. Empirical performance and regulatory approval are not granted.

[1.5in]

validation
Observed binary-outcome validation and exact common-PD count testing

---

Description

Compute weighted discrimination and proper scoring metrics on observed binary outcomes. Test a count under an explicitly independent common-probability binomial model. Applicability and exact-boundary conventions remain visible.

Usage

```
auc_gini(outcomes, scores, weights = NULL, score_direction,
    context = NULL)
ks_discrimination(outcomes, scores, weights = NULL, score_direction,
                  context = NULL)
brier_score(outcomes, predicted_pd, weights = NULL, context = NULL)
log_loss(outcomes, predicted_pd, weights = NULL,
    boundary_policy = "exact",
          epsilon = NULL, context = NULL)
binomial_pd_test(n, observed_defaults, common_pd, alternative,
                 independence_assumption, confidence_level = 0.95,
                 context = NULL)
calibration_intercept_slope(predicted_pd, outcomes, design,
    covariance_policy,
  cluster_ids = NULL, confidence_level = .95,
    boundary_policy = "reject",
  epsilon = NULL, max_iter = 100, tolerance = 1e-10, context = NULL)
calibration_curve(predicted_pd, outcomes, bin_edges, design,
    interval_method,
  dependence_assumption, confidence_level = .95, minimum_count = 5,
  context = NULL)
spiegelhalter_test(predicted_pd, outcomes, design, alternative,
  independence_assumption, context = NULL)
poisson_binomial_pd_test(individual_pd, observed_default_count,
    design,
  alternative, independence_assumption,
    method = "dynamic_programming",
  context = NULL)
hosmer_lemeshow_test(predicted_pd, outcomes, bin_edges, design,
  degrees_of_freedom_policy, independence_assumption,
    minimum_expected = 5,
```

```
  context = NULL)
dependent_pd_backtest(predicted_pd, outcomes, cluster_ids, time_ids,
    design,
  null_model, innovations, alternative, confidence_level = .95,
    context = NULL)
default_count_backtest(observation_panel, frozen_forecasts,
    group_by, unit,
  horizon_days, default_definition, redefault_policy,
    evaluation_cutoff,
  context = NULL)
```

Arguments

`outcomes`

Nonempty finite numeric vector of observed zero/one outcomes. Missing, censored or unfunded outcomes must not be encoded as zero.

`scores`

Finite numeric score vector of the same length as outcomes.

`predicted_pd`

Finite numeric probability vector in [0,1], of the same length as outcomes, matching the observed event and outcome horizon.

`weights`

Optional nonnegative finite vector, same length as outcomes, with positive finite total. Defaults to one per observation. No silent recycling.

`score_direction`

Required higher_is_risk or lower_is_risk; risk predicts outcome one.

`boundary_policy`

For log loss: exact or clip. For diagnostic calibration: reject or clip.

`epsilon`

Required scalar in (0,0.5) for clipping, producing representable interior lower and upper bounds. Must be NULL under exact/reject policy.

`n`

Nonnegative exact integer number of independent trials, no larger than 2^{53} minus one.

`observed_defaults`

Exact integer count from zero through n.

`common_pd`

Common Bernoulli probability under the null, in [0,1].

`alternative`

Required two-sided, less or greater.

`independence_assumption`

independent_common_probability for the binomial test; independent_fixed_bernoulli for Spiegelhalter, Poisson-binomial and HL tests.

`confidence_level`

Scalar strictly between zero and one.

`design`

Named frozen-forecast contract with prediction_origin HOLDOUT/EXTERNAL, forecast_known_at, evaluation_at, split_id and complete physical target. Optional forecast_fingerprint is verified; computed hash is always returned. Dependent tests also require ordered observation_ids. FITTED_SAMPLE is allowed only for its explicit HL design policy.

`covariance_policy`

iid, cluster_cr1 or none for diagnostic logistic fit.

`cluster_ids`, `time_ids`

One nonempty cluster/time identifier per observation. Cluster covariance treats distinct clusters as independent. Simulation maps unique cluster/time pairs to correlated factors.

`max_iter`, `tolerance`

Native GLM iteration and convergence controls.

`bin_edges`

Prespecified strictly increasing numeric edges covering [0,1].

`interval_method`

none or clopper_pearson reliability-bin intervals.

`dependence_assumption`

descriptive_only, or independent_common_probability_within_bins for exact binomial intervals.

`minimum_count`

Positive integer count threshold for descriptive small-bin flag.

`individual_pd`

Numeric vector of independent heterogeneous Bernoulli PDs.

`observed_default_count`

Integer realized count between zero and trial count.

`method`

dynamic_programming exact Bernoulli convolution.

`degrees_of_freedom_policy`

external_common_bin_pd or fitted_logistic_approximation; must match prediction origin.

`minimum_expected`

Positive expected event/non-event count threshold for withholding unreliable chi-square p-values.

`null_model`

Named Gaussian-factor model with family gaussian_factor, asset_correlation scalar/vector, factor_correlation row-list matrix and parameter_uncertainty conditional_fixed or supplied_predictive_pd_draws. The latter requires parameter_draw_source.

`innovations`

Data-only object with draw_assumption independent_standard_normal_rows, ordered observation_ids, sorted factor_keys (cluster_id/time_id objects), factor_normals B-by-G and idiosyncratic_normals B-by-n. Matrices are explicit row lists. Optional parameter_pd_draws is B-by-n.

`observation_panel`

Rows with observation_id, default-unit key, as_of, performance_start, outcome_end, eligible, prior_default, default_definition, explicit target (possibly NULL), outcome_known_at when observed, and grouping fields.

`frozen_forecasts`

Rows with observation_id, scalar pd, forecast_at, known_at, default_definition, horizon_days, unit, probability_measure PHYSICAL, model_id and model_version.

`group_by`

Distinct observation field names defining descriptive cohorts.

`unit`

obligor (entity_id key) or facility (facility_id key).

`horizon_days`

Positive integer calendar-day observation horizon.

`default_definition`

Explicit common event-definition string.

`redefault_policy`

Required exclude_previous_default.

`evaluation_cutoff`

ISO calendar date for available mature outcomes.

`context`

Optional calculation context or context CalcResult.

Details

AUC compares every event/non-event pair. An event with higher oriented risk score is concordant; equal scores receive half credit. Pair weights are products of within-event and within-non-event normalized weights. The algorithm groups tied scores before accumulating lower-score non-event mass. Gini is 2*AUC-1. Zero-weight rows contribute neither pair count nor pair mass, but inputs are still validated. If either class has no positive weight, AUC/Gini and pair masses are NULL with NOT_APPLICABLE and ONE_CLASS. No confidence interval is inferred from these weighted descriptive statistics.

KS is the maximum absolute difference between the weighted non-event and event empirical CDFs of oriented risk score. Ties move together. The returned signed difference is non-event CDF minus event CDF. Equal maximal distances select the first threshold in ascending oriented score order. Under lower_is_risk, original score thresholds consequently appear in descending order. A missing weighted class yields NOT_APPLICABLE, empty cdf and NULL summary values. Weighted, tied or dependent credit observations do not imply the classical continuous IID KS null or a valid classical KS p-value.

The binary Brier convention is the mean squared probability error on [0,1], without multiplication by two. The baseline is the observed weighted event rate times one minus that rate; it is a descriptive in-sample constant-prediction baseline, not an independent model benchmark.

Log loss uses natural logarithms (nats). Correct exact predictions p=0/y=0 and p=1/y=1 have zero loss. An impossible prediction with positive weight has infinite loss, recorded as NULL plus INFINITE rather than a silently clipped number. Its individual contribution is NULL and infinite_loss_ flags entry TRUE. An impossible prediction of zero weight contributes zero and does not make total loss infinite. Clip policy explicitly truncates to [epsilon,1-epsilon]; diagnostics record the epsilon and number of probabilities changed. Clipping changes the score at those boundaries. Very small epsilon values that round the upper bound to one are rejected. Missing/nonfinite inputs are not silently removed.

The binomial test uses native stats::binom.test, mapping the public two-sided spelling to its two.sided alternative. Its two-sided p-value sums outcomes whose null probability mass is no greater than the observed mass, using the numerical tolerance of that implementation. Confidence intervals are exact Clopper-Pearson intervals; one-sided alternatives produce matching one-sided limits. An empty cohort returns NOT_APPLICABLE, EMPTY_COHORT and NULL metrics. The method explicitly requires independent common-probability trials. Correlated obligors or heterogeneous PDs require a different null model; the assertion is supplied by the caller.

All observations must refer to the same intended default definition, target horizon and evaluation population as their predictions. These functions cannot repair selection, leakage, censored targets or wrongly measured outcomes.

Value

CalcResult with named-list value, units, diagnostics, provenance and limitations. Scalar input is one observation; vectors never recycle.

auc_gini returns auc, gini, applicability/reason, pair_count, tied_pair_count, concordant_pair_mass and tied_pair_mass. Pair counts count rows with positive weight; masses use class-normalized pair weights. ks_discrimination returns ks, threshold, signed_difference, full cdf row list and applicability/reason.

brier_score returns brier, individual normalized-weight contributions, unweighted squared_errors, observed_rate and outcome_variance_baseline. log_loss returns log_loss, loss_status, contributions, infinite_loss_flags; positive infinity is represented by NULL plus INFINITE so output remains finite-JSON compatible.

binomial_pd_test returns p_value, lower/upper confidence limits, observed_rate, applicability and reason.

Frozen calibration diagnostics

Extended diagnostics compute a canonical forecast hash and verify an optional expected hash. Forecast known_at must not follow evaluation_at. Design includes split_id, origin and a target with event, positive horizon, time_unit, physical probability_measure and conditioning. These are explicit evidence assertions, not automatic proof of an independent test sample.

calibration_intercept_slope fits logit Pr(Y=1)=intercept+slope*logit(frozen PD). It returns intercept, slope, covariance, standard_errors, confidence_intervals, applicability/reason and converged. It does not apply the fitted coefficients to original predictions. Constant PDs make the joint coefficients unidentified; zero/one PDs yield BOUNDARY_PD unless explicit clipping is chosen. One class, separation, rank failure and nonconvergence return missing estimates with reasons. The native GLM checks identification and likelihood convergence.

IID covariance is inverse expected Fisher information with normal intervals. Cluster CR1 aggregates observation likelihood score vectors within each named cluster, sandwiches their outer products and applies G/(G-1)*(n-1)/(n-2). It requires at least two independent clusters and n>2; intervals use Student t with G-1 degrees of freedom. Original model parameter uncertainty is not included. The none policy returns coefficients without covariance/intervals.

calibration_curve uses left-closed bins, except that the last also includes one. Returned bins retain empty cells and their missing rates/intervals, counts, defaults, mean PD, observed rate, residual, limits, small_bin flag and status. Assignments and totals permit reconciliation. Exact Clopper-Pearson limits require independent_common_probability_within_bins; descriptive_only with no interval is appropriate when that assumption is unsupported. Bins must not be chosen by inspecting the outcomes they evaluate.

Independent fixed-forecast tests

spiegelhalter_test computes numerator sum((y-p)*(1-2*p)) and variance sum(p*(1-p)*(1-2*p)^2), with Z their standardized ratio. It returns numerator, variance, z, normal-approximation p_value and applicability/reason. Zero variance, including all p=0.5, is NOT_APPLICABLE. Maximum single-term variance share is exposed. A small or dominated sum can invalidate normal approximation. This is one calibration moment, not an omnibus test, and one-sided directions refer to this moment.

poisson_binomial_pd_test multiplies Bernoulli probability polynomials by successive native dynamic-programming convolutions. Returned counts, probability_mass, expected_count, variance and inclusive-tail p_value describe independent heterogeneous Bernoulli trials. Two-sided probability-mass ordering uses relative tolerance 1e-7. Equal PDs recover the binomial distribution, and endpoint PDs are deterministic trials. Mass error and zeroed feasible-support cells are reported. Unrepresentable positive tails yield missing p_value and NUMERICAL_UNDERFLOW; a truly impossible count can have exact zero probability. An empty cohort is NOT_APPLICABLE. Work is quadratic in trial count; model estimation uncertainty and dependence are excluded from this conditional null.

hosmer_lemeshow_test reports per-bin (O-E)^2/[E*(1-E/n)], total statistic, degrees_of_freedom, p_value, bin evidence and applicability/reason. External fixed forecasts use external_common_bin_pd, requiring equal individual probabilities within each bin and degrees of freedom equal to nondeterministic occupied bins. The fitted_logistic_approximation policy requires FITTED_SAMPLE and uses G-2 as an explicit conventional approximation, not a universal fitted-model correction. Empty/deterministic bins remain visible. Small expected event/non-event cells, heterogeneous external-bin PDs, impossible deterministic outcomes and nonpositive degrees of freedom withhold a p-value. An impossible cell makes the statistic missing rather than finite. Independence within and between bins remains required.

Declared dependent null simulation

dependent_pd_backtest uses unique ordered observation_ids and sorted distinct (cluster_id,time_id) factor keys. Supplied standard-normal innovation matrices have B>=2 rows. No RNG is called. A Cholesky factor of the positive-definite unit-diagonal factor-correlation matrix induces the declared cross-factor dependence. The latent value for observation i is sqrt(rho_i)*factor_i+sqrt(1-rho_i)*idiosyncratic_i, with default at or below qnorm(p_i). Rho lies in [0,1]; shared keys preserve common shocks and factor correlation can represent time dependence. Perfectly common factors should share a key instead of using a singular correlation matrix.

Under conditional_fixed, probabilities are fixed across paths. Under supplied_predictive_pd_draws, supplied B-by-n probability paths determine the thresholds and parameter_draw_source must explain the predictive ensemble. Independent standard-normal rows and their relationship to parameter draws are caller assertions; the library cannot certify sampling assumptions from realized innovations. A predictive ensemble is not an automatically fitted frequentist parameter-sampling distribution.

The statistic is observed count minus fixed expected count. Inclusive one-sided tails and two-sided absolute count residuals give p=(extreme+1)/(B+1). Output retains all null_counts/null_statistics, observed/expected counts, statistic, p_value, null_count_interval and tail_probability_interval. The count interval uses the empirical inverse CDF (R quantile type 1). The exact binomial tail-probability interval measures Monte Carlo error, not PD uncertainty. Diagnostics disclose finite p-value resolution, factor keys, null model and exact innovation fingerprint.

Mature default-count cohorts

default_count_backtest joins frozen forecasts by observation_id. Creation time must not exceed forecast knowledge, which must not exceed the observation as_of. Both forecast and observation must match the declared event, physical measure, unit and positive calendar-day horizon; performance cannot start before as_of. Observed outcomes require knowledge dates, and a nondefault zero cannot be known before the outcome window completes. Missing forecasts are explicit exclusions.

Obligor policy deduplicates identical entity/as_of/performance-start/end windows across facilities. PD, target, group and eligibility disagreement raises a conflict. Facility policy uses facility identity. The function does not infer an obligor outcome by combining inconsistent facility labels. Previous-default units are excluded by the declared policy. Only mature, complete, known and eligible windows contribute to both observed and expected counts, even if default was observed early. Future, immature, missing and ineligible rows are exposed as exclusions.

Output groups and overall totals give unit_windows (with available forecasts), observed_count, observed_defaults, expected_defaults, O/E ratio/status, observed_rate and mean_pd. Group totals

reconstruct the compared population. Missing forecasts are excluded before unit aggregation. Input/forecast hashes and duplicate-row counts retain evidence. An expected count of zero produces missing O/E with UNDEFINED when observed count is zero, or INFINITE when defaults occurred. Repeated time windows can remain dependent: this report supplies no binomial test.

References

Spiegelhalter (1986), Probabilistic prediction in patient management and clinical trials, doi:10.1002/sim.4780050506. Formula cross-checked against Harrell's primary rms validation implementation at https://github.com/harrelfe/rms/blob/master/R/val.prob.s.

BIS (2005), Studies on the Validation of Internal Rating Systems, Working Paper 14, https://www.bis.org/publications/studies-validation-internal-rating-systems-revised, fixed-bin chi-square construction and caveats.

Phipson and Smyth (2010), Permutation P-values Should Never Be Zero, https://gksmyth.github.io/pubs/PermPValuesPreprint.pdf, Monte Carlo tail-rank principle.

See Also

`stats::binom.test`

Examples

```
auc_gini(c(0, 1, 0, 1), c(0, 1, 1, 2), c(2, 2, 3, 4),
         score_direction = "higher_is_risk")$value$auc
brier_score(c(0, 1), c(0.25, 0.75), c(1, 3))$value$brier
log_loss(c(0, 1), c(0, 0))$value$loss_status
binomial_pd_test(4, 0, 0.5, alternative = "less",
                 independence_assumption = "independent_common_probability")$value
```

[1.5in]

`validation_backtests`
Frozen Loss, Survival, Population and Migration Validation

Description

Dated observed-outcome backtests and descriptive population distances, separating metrics, uncertainty, applicability and decisions. Open workouts and censored endpoints are not fabricated as complete outcomes.

Usage

```
lgd_backtest(records, design, workout_policy, uncertainty,
  cluster_resamples = NULL, confidence_level = .95, context = NULL)
ead_backtest(records, design, weighting_policy, uncertainty,
  cluster_resamples = NULL, confidence_level = .95, context = NULL)
survival_backtest(predicted_event_curves, observations, horizons,
  censoring, design, uncertainty, cluster_resamples = NULL,
  confidence_level = .95, context = NULL)
```

```
population_stability(reference, current, binning, metrics,
    smoothing,
  design, reference_weights = NULL, current_weights = NULL,
    context = NULL)
rating_migration_backtest(records, predicted_transition_matrices,
  states, state_contract, design, uncertainty, weights = NULL,
  cluster_resamples = NULL, confidence_level = .95, context = NULL)
```

Arguments

`records`, `observations`

Dated forecast records with unique prediction ID, entity/cluster/segment, as_of, forecast_known_at, horizon_end, matching model/parameter identities and outcome_known_at. Method-specific fields are described below.

`design`

Model ID/version, parameter version, data fingerprint, population, split, HOLDOUT/EXTERNAL origin, evaluation cutoff and target. Target declares default event, horizon, unit, measure and conditioning.

`workout_policy`

Closed-only with reference, or frozen closure IPCW with reference, identification assumption, model fingerprint/known-at and minimum observation probability.

`weighting_policy`

Method equal, reference_limit or supplied, plus reference. Supplied weights use each row's weight field.

`uncertainty`

None or cluster_bootstrap using supplied shared draws.

`cluster_resamples`

At least two lists/vectors each sampling the original number of known sampling clusters with replacement. Entire clusters move together.

`confidence_level`

Interior confidence level for percentile intervals.

`context`

Optional matching target context; its knowledge date must equal evaluation cutoff.

`predicted_event_curves`

Aligned row-by-horizon nondecreasing cumulative default probabilities.

`horizons`

Increasing positive event-curve times within the target horizon.

`censoring`

Complete-followup contract or frozen marginal IPCW step survival with model provenance, support, positivity threshold, independence and tie convention.

`reference`, `current`

Numeric or categorical population values, with missing values separately identified.

`binning`

Frozen binning ID/version/known-at and numeric cutpoints/reference support or categorical levels.

`metrics`

Any of psi, jsd and numeric wasserstein.

`smoothing`

Explicit nonnegative PSI pseudocount per fixed cell. JSD remains unsmoothed.

`reference_weights`, `current_weights`

Optional aligned nonnegative observation weights.

`predicted_transition_matrices`

One row-stochastic state matrix or one aligned matrix per forecast.

`states`

Distinct ordered state IDs defining matrix rows and columns.

`state_contract`

Scale ID/version/reference/known-at, disjoint default/cure/exit states and explicit absorbing states.

`weights`

Optional aligned nonnegative migration weights.

Details

Outcome backtests require physical target probabilities and frozen predictions. Records must match declared model/parameter identities. Forecast knowledge cannot follow as_of. Loss and migration windows use integer days or whole calendar months with month-end clamping. Immature windows and future/unknown outcomes are excluded with reasons. Bootstrap intervals use type-7 percentiles conditional on frozen prediction and nuisance models; if any draw cannot estimate a metric, its interval is non-estimable instead of discarding that draw. Weight effective sample size measures concentration, not independent sample count. No automatic acceptance threshold or p-value is produced.

LGD conditions on default. Rows add default_date, predicted_lgd, OPEN/CLOSED workout status; completed rows add workout_closed_at, realized_lgd and nonnegative ead_at_default. Default lies between forecast date and horizon end; default-date LGD forecasts are allowed. Closure follows default and precedes outcome knowledge. Open rows never supply final realized LGD. Closed-only metrics explicitly describe a selected completed-workout population. IPCW requires

closure_independent_of_final_lgd_given_recorded_covariates, frozen model fingerprint/known-at before forecasts and a positive interior minimum_observation_probability. Supplied per-row observation_probability lies in [0,1]; below-threshold support is excluded and the estimand is reduced explicitly. Count and EAD weighting are separate. Self-normalized inverse-closure weights produce bias (prediction minus outcome), MAE, RMSE, means and row bias contributions, overall/by segment with corresponding bootstrap intervals. Raw economic LGD is not clipped.

EAD is default-conditioned and requires matching design/row currency, strictly pre-default forecasts, nonnegative predicted/realized EAD, realized_ead_date equal to default_date, and reference drawn amount/limit. Unweighted and selected weights are reported separately. CCF equals (EAD_ default minus drawn_reference) divided by (limit_reference minus drawn_reference). Fully drawn and overlimit lines have undefined CCF, not zero, while remaining EAD observations. No clipping occurs.

Survival conditions on performing. Each observation adds observed_time and Boolean event; FALSE means censoring. The design declares days_per_time_unit: days 1, months 30 or 365/12 or 365.25/12, years 365 or 365.25. These are elapsed-time conventions. Horizon end is as_of plus ceiling of target horizon times that factor; followup cannot be known before its corresponding elapsed date. Complete followup requires every nonevent strictly beyond the final horizon. Frozen IPCW supplies increasing times, nonincreasing survival, support_end, interior minimum_survival, model fingerprint/known-at, event_before_censoring tie rule and censoring_independent_of_event_ features_and_across_clusters assumption. The routine does not fit censoring on test outcomes.

The censoring curve G(t)=P(C>t) is right continuous and one before its first knot. At h, event cases with time<=h use 1/G(time-), controls with observed time>h use 1/G(h). Censored observations at/before h contribute no invented label. Brier divides by the full eligible cohort, not IPCW sum; its finite-sample value can exceed one. Unsupported horizons receive no metric. Dynamic AUC compares cases/controls from different declared clusters, with half credit for score ties. Truncated concordance uses the last-horizon PD score, event time t_i<=last horizon and t_j>t_i in different clusters; event-time ties are excluded. Pair weights are 1/(G(t_i-)*G(t_i)), reducing to Uno 1/G squared under continuous censoring. Integrated Brier uses the supplied-grid trapezoidal average only. The marginal censoring contract cannot repair informative/feature-dependent censoring. Bootstrap holds the censoring model fixed.

Population stability adds reference_as_of/current_as_of to design; bins must be known before both samples and samples before evaluation. Numeric cutpoints assign equality to the lower bin and reserve a missing bin. Categorical bins reserve separate missing and unseen categories. Missingness and novel/out-of-support observations are reported separately. PSI uses explicit fixed-cell pseudocounts; without smoothing, unequal zero/nonzero masses yield null plus INFINITE, while two zeros contribute zero. JSD is unsmoothed symmetric KL-to-mixture in natural logarithms. Wasserstein is the first raw numeric distance excluding and separately reporting missingness. Empty weighted populations are non-estimable. No universal drift threshold or significance claim follows.

Migration conditions on origin_state. States and scale/version are fixed before forecasts; default, cure and exit roles are distinct. Nonnegative row-stochastic matrices are checked within 1e-12 without normalization; declared absorbing rows must be exact identity. Records add matching scale identity, origin state, observed/censored status, destination state and state_observed_at. Nonabsorbing endpoints require exact horizon observation; earlier observations imply endpoints only for declared absorbing states. Censored rows are not self-transitions. Observed departures from an absorbing state are reported as data/definition violations and excluded. Empty origin grades have no probability estimates or perfect goodness. Multiclass Brier sums K squared errors (range 0 to 2); impossible outcomes retain null plus INFINITE log loss. Destination-default share is not first-

default incidence. Observed/expected counts, residuals and cluster-bootstrap transition intervals remain visible.

Value

CalcResult containing a ValidationResult payload with metrics, uncertainty, null p-value, NOT_ASSESSED decision, explicit applicability, design, coverage/exclusion reasons, diagnostics and content fingerprints. Population stability returns analogous descriptive distribution diagnostics.

References

BCBS validation background: https://www.bis.org/publications/studies-validation-internal-rating-systems-revised. Censored scoring documentation: https://scikit-survival.readthedocs.io/en/stable/api/generated/sksurv.metrics.brier_score.html, https://scikit-survival.readthedocs.io/en/stable/api/generated/sksurv.metrics.cumulative_dynamic_auc.html, https://scikit-survival.readthedocs.io/en/stable/api/generated/sksurv.metrics.concordance_index_ipcw.html. Discrete event-left/control-right and cross-cluster conventions are explicit local extensions derived from observation probabilities; no regulatory approval is claimed.

[1.5in]

`validation_benchmarks`
Paired Frozen Predictions and Dated Rating Benchmarks

Description

Compares genuine paired mature prediction cases or explicitly mapped dated local ratings, preserving exclusions and non-estimability.

Usage

```
benchmark_predictions(model_cohorts, reference_model_id, metrics,
 matching_policy, multiple_comparisons, design, uncertainty,
 cluster_resamples = NULL, confidence_level = 0.95, context = NULL)
benchmark_ratings(internal_ratings, benchmark_records,
    scale_mapping,
 matching_policy, design, uncertainty, cluster_resamples = NULL,
 confidence_level = 0.95, context = NULL)
```

Arguments

`model_cohorts`

Named model entries containing sealed monitoring cohort, model and parameter versions, fingerprint and training knowledge date.

`reference_model_id`

Reference model key.

`metrics`

Brier, log loss, AUC, Gini, calibration error or mean PD, using lower-case names with underscores.

`matching_policy`

Prediction strict/intersection policy; for ratings a list specifying provider, exact_date/latest_known method, max_age_days, watch_policy, kappa_weighting and benchmark_origin.

`multiple_comparisons`

none or bonferroni across candidate/metric comparisons.

`design`

Frozen HOLDOUT/EXTERNAL validation model, parameter, data, population and split identifiers, evaluation_cutoff and target contract.

`uncertainty`

none or cluster_bootstrap.

`cluster_resamples`

Shared lists of cluster IDs sampled with replacement, preserving original cluster count.

`confidence_level`

Interior confidence level.

`context`

Optional matching calculation context.

`internal_ratings`

Dated internal records with common prediction identity, dates, model versions and rating entity/unit, scale, status, grade and concept fields.

`benchmark_records`

Local external records with unique benchmark_id, entity/unit/provider, effective and knowledge dates, target, scale, status, grade and rating concepts.

`scale_mapping`

Frozen identified mapping, ordered common grades, identified internal/benchmark scales and complete order-preserving maps to common grades.

Details

Prediction cohorts must agree on case identity, dates, weights and known labels. Negative outcomes require complete-through coverage; maturity precedes all outcome metrics. Differences are candidate minus reference. Calibration error is observed minus predicted. Infinite boundary log loss and

single-class AUC remain explicit. Shared cluster percentile intervals condition on frozen models; undefined replicates suppress intervals. Bonferroni is an approximate bootstrap family adjustment, not correction for selecting candidates.

Rating concepts, targets and units must match exactly; no PIT/TTC, currency, issuer/issue or support conversion is inferred. Latest known matching chooses a record before testing withdrawal, watch, age and scale, without falling back to an earlier active record. Ambiguous equal-date records error. Signed notches are positive for worse internal ratings. Weighted Cohen kappa uses empirical margins; Spearman uses tied midranks. Degenerate margins are non-estimable. Derived mappings are not independent outcome calibration evidence. Rating agreement does not require outcome maturity. Declared provenance is not authenticated, and observed performance differences are not causal effects.

Value

CalcResult with ValidationResult metrics, uncertainty, applicability, explicit coverage, matching diagnostics, frozen design, fingerprints and NOT_ASSESSED decision.

References

Cohen (1968), Psychological Bulletin 70, 213–220, doi:10.1037/h0026256.

[1.5in]

`validation_interventions`
Dated Override and Separately Observed Stage Validation

Description

Replays active override chains and validates separately observable default, transfer, joint and cashflow stages.

Usage

```
override_monitoring(pre_override, post_override, override_ledger,
    policy,
 design, metrics, uncertainty, cluster_resamples = NULL,
 confidence_level = 0.95, context = NULL)
support_transfer_backtest(records, stage_contract, union_contract,
 design, uncertainty, cluster_resamples = NULL,
 confidence_level = 0.95, context = NULL)
```

Arguments

`pre_override`, `post_override`

Distinct model entries in the paired benchmark format, containing sealed monitoring cohorts and frozen version/knowledge metadata.

`override_ledger`

Unique override_id records with case_id, reason, authority, applied_at, known_at, expires_at, from_pd and to_pd.

`policy`

Reference, approved_authorities and explicit concentration_threshold.

`design`

Frozen validation identities, target, HOLDOUT/EXTERNAL origin and evaluation cutoff; currency for amount stages.

`metrics`

Paired prediction metrics accepted by benchmark_predictions.

`uncertainty`

none or cluster_bootstrap.

`cluster_resamples`

Actual shared cluster IDs sampled with replacement.

`confidence_level`

Interior confidence level.

`context`

Optional matching context.

`records`

Common dated validation records with stage_predictions, stage_outcomes and optional nonnegative weight.

`stage_contract`

Named stages with binary/amount kind, observable Boolean, event_definition and reference; amount stages also currency.

`union_contract`

Four distinct binary names: component_a, component_b, joint and union.

Details

An override is active if applied_at is no later than forecast, forecast precedes expires_at and known_at is no later than forecast. Active approved actions must chain exactly from pre-PD to post-PD. Expired/future-known and multiple actions are separately reported. Equal-date actions require upstream resolution. Rates cover all paired forecasts; performance covers common mature eligible cases. Changes are descriptive, not causal.

Observed stages provide value, known_at, complete_through and event_ids. Binary events also require aligned event_times inside the forecast horizon; shared IDs have consistent dates. Negative

binary labels have empty event arrays. Every observed stage requires mature complete follow-up, and stage-specific missingness remains explicit. Unobservable counterfactual stages cannot carry observed labels. Amounts are nonnegative with declared common currency. Predicted joint and union obey Frechet and inclusion-exclusion; observed four-stage labels reconcile only when separately available. Union source IDs deduplicate component events. Missing labels are never reconstructed from final defaults. Outputs give weighted bias, MAE/RMSE, binary Brier and genuine cluster intervals; undefined resamples suppress intervals. No correction for informative stage missingness or counterfactual support identification is implied.

Value

CalcResult with ValidationResult metrics, stage or ledger traces, coverage, applicability, actual uncertainty and NOT_ASSESSED decision.

[1.5in]

`validation_numerics`
Simulation Precision and Evaluated Frozen Model Sensitivity

Description

Separates genuine sampling uncertainty from paired grid changes and evaluates frozen models under explicit numerical input or graph-parameter interventions.

Usage

```
simulation_convergence(runs, shared_innovations, sampling_contract,
 precision, design, context = NULL)
sensitivity_analysis(model, baseline, perturbations, method,
 shared_samples, design, context = NULL)
```

Arguments

`runs`

Path results with replicate_id, time_step, sample_size, ordered path_ids, values and canonical innovation_fingerprint.

`shared_innovations`

Named replicates with ordered path_ids and path-by-innovation values.

`sampling_contract`

iid_paths_with_independent_replicates sampling_unit, mean/bernoulli_probability metric, reference, finest_time_step and time_unit; optional analytic_reference value/source.

`precision`

confidence_level, mc_half_width and grid_difference_tolerance.

`design`

Frozen validation identities, target and evaluation cutoff.

`context`

Optional matching calculation context.

`model`

Evaluator descriptor: pd_model kind with fitted_model, or model_graph kind with model_graph, input_data, parameter_snapshots, execution_context and zero-based output_row.

`baseline`

Numeric coordinate vector.

`perturbations`

Named coordinates with location, lower, upper and step. PD coordinates use feature location and full trained order. Graph coordinates use input/parameter location and a mixed string/integer path; numeric indices are zero based in both languages.

`method`

finite_difference or variance_based.

`shared_samples`

NULL for local derivatives; global list of A/B matrices and distribution_contract with independent_ inputs TRUE and reference.

Details

Simulation levels require exact shared path prefixes and declared fingerprint matches, at least two paths, nested step multiples and all declared replicates at every level. Fingerprints do not authenticate external simulator execution. Means use sample standard deviation over square root of path count; normal mean intervals are asymptotic. Bernoulli Wilson intervals retain positive upper uncertainty at zero defaults. Paired coarse-minus-fine differences use the same paths. Threshold flags are diagnostic and cannot bound continuous-time discretization bias. Intervals are unadjusted.

Sensitivity evaluates actual predict_pd or evaluate_model_graph outputs. Graph interventions may change numeric input value leaves or declared node parameter leaves, preserving topology/contracts/frozen fitted snapshots. Baselines must equal those leaves. Central or one-sided second-order differences respect bounds and report step-halving differences; elasticity is undefined at zero output. Unrepresentable steps error. Shared A/B designs generate one- and two-coordinate hybrids for Jansen first, total and pair interaction indices. Pooled population variance is the denominator. Correlated coordinate declarations are rejected, zero output variance is non-estimable, finite-sample indices are not clipped and global uncertainty is explicitly not estimated. Independence is declared, not authenticated; neither numerical sensitivity nor variance decomposition is causal.

Value

CalcResult with ValidationResult metrics, explicit diagnostics, evaluated coordinate traces or innovation nesting, fingerprints and DIAGNOSTIC_ONLY decision.

References

Jansen (1999), Computer Physics Communications 117, 35–43, doi:10.1016/S0010-4655(98)00154-4. NIST/SEMATECH e-Handbook, section 7.2.4.1, Wilson confidence intervals.

[1.5in]

`validation_retail`
Retail Cohort and Exact Monthly Transition Validation

Description

Audits supplied retail population/target tables without rebuilding targets, and scores exact-horizon state transitions using shared migration validation.

Usage

```
validate_retail_cohort(application_targets, account_panel,
 behavioral_targets, target_profile, split_memberships,
 source_lineage, design, context = NULL)
roll_rate_backtest(forecasts, account_panel, states, state_contract,
 design, weighting, uncertainty, cluster_resamples = NULL,
 confidence_level = 0.95, context = NULL)
```

Arguments

`application_targets`

Existing operational application-target rows, preserving approved/accepted/booked/funded distinctions.

`account_panel`

Existing account-month snapshots for cohort audit; observed dated state endpoints with explicit risk/default episode IDs for transitions.

`behavioral_targets`

Existing account/date behavioral targets and follow-up/maturity fields.

`target_profile`

Frozen profile ID/version/reference/known_at; default definition/unit/group rule, distinct event concepts, first-default/exit conventions, application-day and behavioral-month horizons, product risk-set rules and split isolation keys.

`split_memberships`

Customer/role rows with all declared isolation-group identifiers.

`source_lineage`

Source ID/hash/kind and observed/available dates per application:id or account:id@date prediction key.

`design`

Frozen validation identities, target and evaluation cutoff.

`context`

Optional matching calculation context.

`forecasts`

Dated frozen transition matrices with common validation fields, account and origin state, risk/default episode IDs and optional opening_balance.

`states`

Ordered state names.

`state_contract`

Versioned known state scale, default/cure/exit/absorbing roles, extended with chargeoff_states subset of default states.

`weighting`

count or opening_balance.

`uncertainty`

none or cluster_bootstrap.

`cluster_resamples`

Actual shared sampling-cluster draws.

`confidence_level`

Interior confidence level.

Details

Cohort audit reports impossible selection flows, outcomes on unfunded contracts, unsupported non-defaults, missing default provenance, calendar/MOB and risk-set conflicts, missing targets, group split overlap and historical feature lookahead. Product-qualified active revolving zero balances remain at risk. No-hit/thin-file/pseudolabel origins are not observed outcomes. Early defaults remain visible but do not bypass mature-performance coverage. Household isolation does not imply shared default. Scope-qualified PASS/FAIL is a supplied-contract audit, not empirical or regulatory approval.

Transition knowledge cannot precede actual state observation; observed endpoints must be exact whole-month horizons except declared absorption. Lagged feature estimates are not observed endpoint states. The historically known origin must match the forecast. Later known revisions use the evaluation cutoff. Missing months never become self-transitions or shorten the horizon. Default-to-cure requires a new risk episode; re-default requires a new default episode. Continuous default cannot silently change episode IDs; absorption departures are violations. Violating paths are excluded with explicit diagnostics. Multiclass Brier/log loss and shared bootstrap use rating_migration_backtest. Calibration conditions on valid observed successors; no informative-missingness correction is inferred.

Value

CalcResult containing ValidationResult metrics, coverage, violations or transition/episode traces, frozen contracts and fingerprints.

[1.5in]

`vintage_backtest`
Fixed Origination Vintage Backtest

Description

Reconciles supplied operational vintage cells with fixed original membership, frozen account probabilities and independently supplied observed account outcomes.

Usage

```
vintage_backtest(vintage_table, account_predictions,
    account_outcomes,
 origin_cohorts, denominator_contract, design, uncertainty,
 cluster_resamples = NULL, confidence_level = 0.95, context = NULL)
```

Arguments

`vintage_table`

Existing operational vintage cells, for example build_vintage_performance output.

`account_predictions`

Account/MOB frozen opening-date forecasts with model/version/parameter IDs and cumulative_pd/interval_pd.

`account_outcomes`

Separate account/MOB dated observed targets with period_end, default_definition, complete/exited flags, cumulative_default and interval_default. Events require event_id and first_event_at; complete no-event observations require complete_through.

`origin_cohorts`

Unique account ID, opening/knowledge dates, original_balance, currency and cluster_id membership.

`denominator_contract`

fixed_origination method, count/original_balance weighting, first_competing_exit treatment, reference/default_definition and origination_month_end_plus_mob horizon convention.

`design`

Frozen physical performing-at-origin validation target; whole-month horizon gives maximum MOB.

`uncertainty`

none or cluster_bootstrap.

`cluster_resamples`

Shared original sampling-cluster draws, preserving repeated accounts across MOBs.

`confidence_level`

Interior confidence level.

`context`

Optional matching context.

Details

Exits never shrink original denominators. Counts, original balances, default/incident/exit totals and complete follow-up must reconcile to operational cells. No substitute target builder is introduced. MOB uses cohort month-end, not equal elapsed days from every opening; MOB zero is the opening month's end. Cumulative forecasts must increase and consecutive interval forecasts equal their increments. Positive/exit events must be observed within the origin-to-end window; interval defaults belong to the named calendar month. Negative completed paths require coverage through the endpoint. Future-known evidence does not enter observed counts.

A cell is estimable only when calendar-mature and every original account has complete follow-up. Incomplete cells retain event counts and visible missing accounts but no fabricated rates or survivor reweighting. Metrics include cumulative and unconditional interval Brier, bias, MAE/RMSE and weighted means. Interval denominators are original cohorts, not survivor hazards. Original-balance weighting is monetary incidence, not account PD. Actual shared cluster resampling supplies conditional intervals; undefined replicates suppress intervals. No censoring adjustment or empirical model approval is inferred.

Value

CalcResult with ValidationResult cell metrics, maturity, original denominators, remaining-first-event risk counts, explicit uncertainty and fingerprints.

Index

affordability, 4
affordability_stress_backtest, 7
aggregate_household_obligations (retail_features), 174
aggregate_netting_set_exposure (ead_counterparty), 82
aggregate_portfolio_loss (loss_paths), 122
allocate_expected_shortfall (credit_loss_distribution), 68
allocate_guarantee_coverage (credit_links), 64
application_risk, 9
apply_assessment_event (rating_system), 161
apply_calibrator (calibrate_intercept), 21
apply_cash_waterfall (simulation_cashflows), 204
apply_credit_decision_policy (compile_credit_decision_policy), 49
apply_guarantee_risk (credit_links), 64
apply_override (credit_links), 64
apply_parent_support (credit_links), 64
apply_qualitative_adjustments (scorecard), 186
apply_rating_constraints (credit_links), 64
apply_satellite_model (macro), 126
assemble_joint_application (application_risk), 9
assess_affordability (affordability), 4
assess_counterfactual_robustness, 12
assess_expert_rating_agreement (expert_probability), 93
assess_policy_robustness (decisioning_policy_evaluation), 76
assess_rating_applicability (rating_system), 161
assess_transfer_applicability (transfer_events), 224
asset_pool_backtest, 13
assign_default_groups (credit_links), 64
assign_delinquency_state (behavioral), 15
assign_model_role (monitoring), 145
assign_retail_risk_pool (retail_pools), 180
asymptotic_factor_loss (credit_loss_distribution), 68
auc_gini (validation), 232

behavioral, 15
benchmark_predictions (validation_benchmarks), 243
benchmark_ratings (validation_benchmarks), 243
binomial_pd_test (validation), 232
block_bootstrap_scenarios (macro), 126
bootstrap_cds_hazard_curve (market_implied), 133
brier_score (validation), 232
build_account_month_panel (behavioral), 15
build_application_snapshot (application_risk), 9
build_application_targets (application_risk), 9
build_behavioral_targets (behavioral), 15
build_bnpl_obligation_features (retail_obligations), 176
build_bureau_features (retail_evidence), 171
build_calculation_context, 20
build_collections_action_history (collections), 42
build_collections_episodes (collections), 42

build_credit_asset_pool (structured_credit_pool_foundations), 213
build_credit_quote_snapshot (market_implied), 133
build_cure_targets (collections), 42
build_customer_household_graph (retail_features), 174
build_debt_schedule (simulation_cashflows), 204
build_distributed_lags (features), 102
build_entity_graph (credit_links), 64
build_financial_ratios (features), 102
build_monitoring_cohorts (monitoring_metrics), 146
build_observation_windows (credit_data), 58
build_payment_history_features (retail_behavior), 168
build_rating_subjects (rating_system), 161
build_recovery_episodes (credit_data), 58
build_retail_sme_features (retail_sme), 183
build_revolving_features (retail_behavior), 168
build_selection_dataset (selection_bias), 189
build_thin_file_features (retail_evidence), 171
build_transaction_features (retail_cashflows), 170
build_vintage_performance (behavioral), 15

calibrate_beta (calibration_models), 24
calibrate_bond_hazard_curve (market_implied), 133
calibrate_distance_pd_mapping (structural), 209
calibrate_grade_pd (calibration_models), 24
calibrate_intercept, 21
calibrate_isotonic (calibrate_intercept), 21
calibrate_lgd_level (calibration_models), 24
calibrate_logistic (calibrate_intercept), 21
calibration_curve (validation), 232
calibration_intercept_slope (validation), 232
calibration_models, 24
cds_expected_cashflow_legs (market_implied), 133
check_assessment_freshness (rating_system), 161
check_expert_preference_consistency (expert_probability), 93
classification_bayesian, 31
classification_flexible, 33
classification_models, 36
classification_neural, 40
classify_monitoring_findings (monitoring), 145
closeout_exposure_at_default (ead_counterparty), 82
collections, 42
collections_backtest, 47
combine_default_transfer (transfer_events), 224
combine_pd_signals (hybrid), 109
combined_loan_to_value (affordability), 4
compare_cds_bond_credit_basis (market_implied), 133
compare_challenger_cohorts (monitoring_metrics), 146
compare_offer_risk (application_risk), 9
compile_credit_decision_policy, 49
compile_discrete_bayesian_network (expert_probability), 93
compile_qualitative_rubric (expert_qualitative), 97
compose_risk_assessment (rating_system), 161
compute_credit_ale, 51
compute_credit_ice, 52
compute_days_past_due (behavioral), 15
compute_limit_frontier (decisioning_offer_economics), 74
compute_risk_based_price_grid (decisioning_offer_economics), 74
compute_woe_iv (features), 102
condition_macro_paths (macro), 126
conditional_pd_from_cumulative (pd_

term_structure_extended), 151
content_fingerprint, 54
counterparty_expected_loss (loss_paths), 122
counterparty_exposure_backtest, 55
covered_bond_dual_recourse, 56
credit_concentration_measures (credit_loss_distribution), 68
credit_data, 58
credit_dependence, 61
credit_links, 64
credit_loss_distribution, 68
credit_permutation_importance, 70
cumulative_pd_from_conditional, 71

debt_service_to_income (affordability), 4
debt_to_income (affordability), 4
decision_policy_backtest, 72
decisioning_offer_economics, 74
decisioning_policy_evaluation, 76
decompose_hybrid_prediction (hybrid), 109
decompose_policy_decision (compile_credit_decision_policy), 49
default_count_backtest (validation), 232
dependent_pd_backtest (validation), 232
detect_cashflow_default, 79
detect_credit_feature_drift (monitoring_metrics), 146
detect_pd_calibration_drift (monitoring_metrics), 146
diagnose_fairness_slices (monitoring_metrics), 146
diagnose_market_credit_identifiability (market_implied), 133
discount_recoveries (lgd), 113
distance_to_default (structural), 209

ead, 80
ead_backtest (validation_backtests), 239
ead_counterparty, 82
ead_models, 85
estimate_recurring_expenses (affordability), 4
estimate_credit_shapley_values (explainability_attribution), 99
estimate_downturn_lgd (lgd_downturn), 115
estimate_long_run_pd, 87
estimate_margin_of_conservatism (calibration_models), 24
estimate_multistate_transitions (survival_models), 215
estimate_open_workout_lgd (lgd_workout), 119
estimate_pool_risk_parameters (retail_pools), 180
estimate_population_risk_aipw (selection_bias), 189
estimate_retail_roll_rates (behavioral), 15
estimate_simulation_risk, 89
estimate_survival_curve (survival_models), 215
estimate_tranche_credit_metrics, 90
estimate_transition_matrix (pd_term_structure_extended), 151
evaluate_decision_grid (decisioning_policy_evaluation), 76
evaluate_model_graph (hybrid), 109
evaluate_qualitative_evidence (expert_qualitative), 97
evaluate_treatment_candidates (decisioning_offer_economics), 74
event_conditioned_loss (loss_paths), 122
expected_collections_value (collections), 42
expected_loss, 92
expected_loss_from_paths (loss_paths), 122
expert_probability, 93
expert_qualitative, 97
explain_linear_credit_prediction (explainability_attribution), 99
explainability_attribution, 99
exposure_schedule (ead), 80

features, 102
find_feasible_credit_counterfactual, 104
first_passage_pd (structural), 209
fit_aft_model (survival_models), 215

fit_bayesian_pd (classification_bayesian), 31
fit_beta_lgd (lgd_models), 116
fit_binary_glm, 106
fit_cause_specific_hazards (survival_models), 215
fit_copula (credit_dependence), 61
fit_cox_ph (survival_models), 215
fit_discrete_hazard (survival_models), 215
fit_discriminant_pd (classification_models), 36
fit_dynamic_factor (macro), 126
fit_ead_model (ead_models), 85
fit_elicited_beta_prior (expert_probability), 93
fit_fine_gray (survival_models), 215
fit_firth_logit (classification_models), 36
fit_fractional_lgd (lgd_models), 116
fit_gam_pd (classification_flexible), 33
fit_gradient_boosting_pd (classification_flexible), 33
fit_hierarchical_calibration (calibration_models), 24
fit_joint_selection_model (selection_bias), 189
fit_market_to_physical_pd_bridge (market_implied), 133
fit_model_selection_protocol (calibration_models), 24
fit_monotone_bins (features), 102
fit_naive_bayes_pd (classification_models), 36
fit_neural_pd (classification_neural), 40
fit_ordinal_model (classification_models), 36
fit_penalized_glm (classification_models), 36
fit_preprocessor (features), 102
fit_random_forest_pd (classification_flexible), 33
fit_recovery_hurdle (lgd_workout), 119
fit_regime_switching (macro), 126
fit_regularization_path (calibration_models), 24
fit_retail_risk_pools (retail_pools), 180
fit_satellite_model (macro), 126
fit_scorecard (scorecard), 186
fit_selection_propensity (selection_bias), 189
fit_sequence_pd (classification_neural), 40
fit_stacked_pd (hybrid), 109
fit_stepwise (calibration_models), 24
fit_svm_score (classification_models), 36
fit_var (macro), 126
fit_vecm (macro), 126
fit_zero_one_inflated_lgd (lgd_models), 116
fuzzy_reject_augmentation (selection_bias), 189

generate_asset_pool_reference_world (synthetic), 221
generate_challenge_cases (synthetic), 221
generate_facility_workouts (synthetic), 221
generate_macro_reference_world (synthetic), 221
generate_obligor_panel (synthetic), 221
generate_retail_reference_world (synthetic), 221
generate_system_event_reference_world (synthetic), 221

hazard_to_pd (pd_term_structure_extended), 151
hosmer_lemeshow_test (validation), 232
hybrid, 109

importance_sample_defaults, 112
incremental_credit_risk (credit_loss_distribution), 68
infer_discrete_credit_network (expert_probability), 93
infer_merton_assets (structural), 209

join_as_of (credit_data), 58
joint_default_probability (credit_dependence), 61

ks_discrimination (validation), 232

lgd, 113
lgd_backtest (validation_backtests), 239
lgd_downturn, 115
lgd_models, 116
lgd_workout, 119
lifetime_expected_loss (loss_paths), 122
load_fitted_model (serialize_fitted_model), 196
log_loss (validation), 232
loss_paths, 122
low_default_bound (calibration_models), 24

macro, 126
make_grouped_temporal_splits (credit_data), 58
make_shared_innovations (credit_dependence), 61
map_pd_to_grade (validate_rating_scale), 227
marginal_pd_from_conditional (cumulative_pd_from_conditional), 71
market_implied, 133
merton_default_probability (structural), 209
merton_equity_value (structural), 209
monitor_rating_cohort (monitoring_metrics), 146
monitoring, 145
monitoring_metrics, 146

normalize_income_period (affordability), 4

override_monitoring (validation_interventions), 245

parcel_rejected_applications (selection_bias), 189
pd_term_structure_extended, 151
pit_ttc_transform, 153
poisson_binomial_pd_test (validation), 232
pool_expert_probability_distributions (expert_probability), 93
population_reweighting (calibration_models), 24
population_stability (validation_backtests), 239
portfolio_simulation, 155
predict_application_risk (application_risk), 9
predict_behavioral_risk (behavioral), 15
predict_cure_and_redefault, 158
predict_ead (ead_models), 85
predict_event_curves (survival_models), 215
predict_lgd (lgd_models), 116
predict_model_score (classification_models), 36
predict_pd (fit_binary_glm), 106
prepare_credit_portfolio (portfolio_simulation), 155
prepare_reject_evaluation (selection_bias), 189
project_retail_roll_rates (behavioral), 15
project_transition_matrix (pd_term_structure_extended), 151

rank_credit_reason_codes, 160
rate_subject_batch (rating_system), 161
rating_migration_backtest (validation_backtests), 239
rating_system, 161
read_local_dataset (credit_data), 58
realized_ccf (ead), 80
receivables_borrowing_base (structured_credit_pool_foundations), 213
reconcile_card_authorizations, 166
reconcile_income_evidence (retail_evidence), 171
register_rating_model_version (rating_system), 161
reject_inference_bounds (selection_bias), 189
reject_inference_sensitivity (selection_bias), 189
release_candidate_assessment (validate_system_coverage), 230
replay_assessment_history (rating_system), 161
replay_consistency (validate_source_lineage), 228
resample_fit_uncertainty (calibration_models), 24
residual_income (affordability), 4

resolve_parent_links (credit_links), 64
retail_behavior, 168
retail_cashflows, 170
retail_evidence, 171
retail_features, 174
retail_obligations, 176
retail_pool_diagnostics, 178
retail_pools, 180
retail_sme, 183
reverse_stress_scenario (macro), 126
risky_bond_expected_cashflows (market_implied), 133
roll_rate_backtest (validation_retail), 249
rollback_model_assignment (monitoring), 145
route_rating_profile (rating_system), 161
run_tranche_waterfall, 185

scale_score_points (scorecard), 186
score_scorecard (scorecard), 186
scorecard, 186
scorecard_reason_codes (scorecard), 186
select_constrained_policy (decisioning_policy_evaluation), 76
select_effective_model (rating_system), 161
selection_bias, 189
selection_bias_sensitivity, 194
selection_ipw_weights (selection_bias), 189
sensitivity_analysis (validation_numerics), 247
serialize_fitted_model, 196
simulate_asset_pool, 197
simulate_asset_values (simulation_drivers), 206
simulate_collateral_margin (ead_counterparty), 82
simulate_contract_counterparties, 198
simulate_copula (credit_dependence), 61
simulate_credit_portfolio (portfolio_simulation), 155
simulate_macro_scenarios (macro), 126
simulate_operating_cashflows (simulation_drivers), 206
simulate_portfolio_migrations (portfolio_simulation), 155
simulate_recovery_waterfall, 200
simulate_repayment_arrangement (collections), 42
simulate_structural_paths (structural), 209
simulate_trade_receivables, 201
simulate_transactions, 202
simulation_cashflows, 204
simulation_convergence (validation_numerics), 247
simulation_drivers, 206
solve_contagion_network (credit_links), 64
spiegelhalter_test (validation), 232
stress_affordability (affordability), 4
stress_pool_counterparties, 207
stress_portfolio_factor_structure (portfolio_simulation), 155
stress_tranche_structure (estimate_tranche_credit_metrics), 90
structural, 209
structured_credit_pool_foundations, 213
support_transfer_backtest (validation_interventions), 245
survival_backtest (validation_backtests), 239
survival_models, 215
synthetic, 221

tranche_loss_from_pool (structured_credit_pool_foundations), 213
transfer_event_default_probability (transfer_events), 224
transfer_events, 224
transfer_loss_components (transfer_events), 224
transform_features (features), 102
transform_macro_history (macro), 126
transition_assessment_state (rating_system), 161

update_expert_beta_binomial (expert_probability), 93
update_odds_from_evidence (expert_probability), 93

validate_credit_dataset (credit_data), 58
validate_dependence_matrix (credit_dependence), 61
validate_probability, 226
validate_rating_scale, 227
validate_retail_cohort (validation_retail), 249
validate_retail_dataset (retail_features), 174
validate_source_lineage, 228
validate_synthetic_world (synthetic), 221
validate_system_coverage, 230
validation, 232
validation_backtests, 239
validation_benchmarks, 243
validation_interventions, 245
validation_numerics, 247
validation_report (validate_system_coverage), 230
validation_retail, 249
vintage_backtest, 251

workout_lgd (lgd), 113