Package: RMTIF (via r-universe)

November 22, 2026

Type Package

Title Restricted Mean Time in Favor of Treatment

Version 0.3.1

Description Estimates restricted mean time in favor of treatment from ordered multistate trajectories, with lower states denoting better outcomes. Provides complete-path pairwise estimation, projection standard errors, normal confidence intervals and tests, terminal-aware inverse probability of censoring weighting, stratified bootstrap inference, and pairwise regression with sandwich or refitted-bootstrap inference. Includes parameterized simulation and Monte Carlo summary tools for reversible processes, continuous-time Markov paths, constant-utility regression, and comparisons of longitudinal oncology endpoints. The restricted-time estimand follows Mao (2023) <doi:10.1111/biom.13570>; pairwise regression is motivated by the proportional win-fractions framework of Mao and Wang (2021) <doi:10.1111/biom.13382>.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, utils, parallel

Suggests MASS, survival, Wcompo

NeedsCompilation no

Author Mohsen Rezapour [aut, cre]

Maintainer Mohsen Rezapour <mrezapou@amgen.com>

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

Date/Publication 2026-11-21 19:10:09 UTC

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

RemoteRef HEAD

RemoteSha 613ac2cc78c4e5297491d6a30399201a6a770cbe

Contents

RMTIF-package . . . 2
rmtif_censoring_support . . . 3
rmtif_complete . . . 4
rmtif_complete_curve . . . 4
rmtif_complete_state_count_curve . . . 5
rmtif_construct_scheduled_visit_states . . . 6
rmtif_ctmc_generator . . . 6
rmtif_gumbel_recovery_study . . . 7
rmtif_hfmock_analysis . . . 8
rmtif_hfmock_subjects . . . 9
rmtif_ipcw . . . 10
rmtif_ipcw_equivalence_check . . . 11
rmtif_ipcw_simulation_parameters . . . 11
rmtif_ipcw_simulation_study . . . 12
rmtif_monte_carlo_summary . . . 13
rmtif_observe_ctmc_path . . . 14
rmtif_oncology_compare_trial . . . 15
rmtif_oncology_design . . . 16
rmtif_oncology_study . . . 17
rmtif_oncology_transition_rates . . . 18
rmtif_pair_times . . . 19
rmtif_poisson_update_generator . . . 20
rmtif_poisson_update_study . . . 21
rmtif_projection_inference . . . 22
rmtif_regression . . . 22
rmtif_simulate_ctmc_path . . . 23
rmtif_simulate_gumbel_regression . . . 24
rmtif_simulate_gumbel_trial . . . 25
rmtif_simulate_ipcw_trial . . . 26
rmtif_simulate_oncology_trial . . . 26
rmtif_simulate_poisson_updates . . . 28
rmtif_state_at . . . 29
rmtif_subject . . . 29
rmtif_true_ctmc_contrast . . . 30

Index **32**

`RMTIF-package` *Restricted Mean Time in Favor of Treatment*

Description

Estimation, inference, and parameterized simulation for ordered multistate trajectories. Lower numeric state values mean better outcomes; net time equals treatment win time minus treatment loss time.

Details

Use `rmtif_subject` to represent observed trajectories, `rmtif_complete` for complete-path inference, `rmtif_ipcw` for censoring-adjusted inference with a refitted stratified bootstrap, and `rmtif_regression` for pairwise regression. Design settings and Monte Carlo sizes are supplied explicitly by the caller.

References

Mao, L. (2023). "On restricted mean time in favor of treatment." Biometrics, 79(1), 61-72. doi:10.1111/biom.13570.

Mao, L. and Wang, T. (2021). "A class of proportional win-fractions regression models for composite outcomes." Biometrics, 77(4), 1265-1275. doi:10.1111/biom.13382.

See Also

`rmtif_subject`, `rmtif_complete`, `rmtif_ipcw`, `rmtif_regression`

`rmtif_censoring_support` *Audit arm-specific censoring support for IPCW*

Description

The aim is to check the empirical positivity condition before estimation. Survival is evaluated as the predictable left limit at ‘tau‘; therefore a Kaplan–Meier curve that first reaches zero exactly at ‘tau‘ still supports integration through that endpoint, whereas a zero before ‘tau‘ does not.

Usage

```
rmtif_censoring_support(subjects,
    tau)
```

Arguments

`subjects`	List of ‘rmtif_subject‘ objects.
`tau`	Candidate restriction horizon.

Value

One row per arm with the left-limit survival at ‘tau‘, the first zero observed by ‘tau‘, maximum empirically verified horizon, and a support flag. Absence of a zero by ‘tau‘ is not extrapolated beyond ‘tau‘.

rmtif_complete *Estimate complete-data rmt-IF with projection inference*

Description

Average pairwise treatment-minus-control net times over every control and treatment subject pair. Subject-level row and column projections yield the two-sample U-statistic standard error. The function reports a normal confidence interval and the z statistic and two-sided p-value for a zero contrast; no independent-increment assumption across time is used.

Usage

```
rmtif_complete(subjects,
    tau,
    conf_level)
```

Arguments

subjects	List of ‘rmtif_subject‘ objects with complete paths to ‘tau‘.
tau	Restriction horizon.
conf_level	Confidence level.

Value

An ‘rmtif_fit‘ list containing ‘estimate‘, projection ‘se‘, ‘conf_int‘, zero-contrast ‘statistic‘ and ‘p_value‘, mean pairwise ‘win‘ and ‘loss‘ times, and row and column projections.

Examples

```
subjects <- list(
  rmtif_subject(0, 0, 5, tau = 3.7),
  rmtif_subject(0, 0, 5, tau = 3.7),
  rmtif_subject(1, 0, 4, tau = 3.7),
  rmtif_subject(1, 0, 4, tau = 3.7)
)
rmtif_complete(subjects, tau = 3.7, conf_level = 0.95)
```

rmtif_complete_curve *Estimate a complete-data rmt-IF curve*

Description

The function generates pointwise estimates and projection intervals for complete ordered-state paths.

Usage

```
rmtif_complete_curve(subjects,
    times,
    conf_level)
```

Arguments

subjects	Complete observed paths.
times	Increasing restriction horizons.
conf_level	Pointwise confidence level.

Value

Data frame with estimate, standard error, interval, and p-value.

---

rmtif_complete_state_count_curve

Evaluate the exact finite-state projection curve by state counts

---

Description

This is the fast implementation of the complete-path pairwise estimator. For each interval it obtains row and column projections from strict empirical state counts in the opposite arm, algebraically matching the dense pair matrix without assuming independent increments across time.

Usage

```
rmtif_complete_state_count_curve(subjects,
    times,
    conf_level)
```

Arguments

subjects	Complete observed ordered-state paths.
times	Increasing positive restriction horizons.
conf_level	Pointwise normal confidence level.

Value

Data frame with horizon, rmt-IF estimate, projection SE, interval endpoints, and two-sided zero-contrast p-value.

`rmtif_construct_scheduled_visit_states` *Construct scheduled-visit ordered states from recurrent HF event records*

Description

This entry point accepts the event-record schema used by the public HF example: patient identifier, event time, raw event code, treatment arm, and baseline etiology. Hospitalization since the previous scheduled visit defines the nonterminal state; observed death is an absorbing exact event.

Usage

```
rmtif_construct_scheduled_visit_states(data,
    tau,
    visit_interval,
    raw_status,
    ordered_state,
    quantile_probs)
```

Arguments

`data`	Event records with ‘id‘, ‘time‘, ‘status‘, ‘Training‘, and ‘HF.etiology‘ columns.
`tau`	Restriction horizon.
`visit_interval`	Scheduled-visit spacing.
`raw_status`	Named ‘censor‘, ‘death‘, ‘hospitalization‘ code vector.
`ordered_state`	Named ordered-state code vector.
`quantile_probs`	Three probabilities for the censor-time arm audit.

Value

List with constructed ‘rmtif_subject‘ paths and subject and arm audit data frames.

`rmtif_ctmc_generator` *Build a continuous-time Markov generator from transition intensities*

Description

The caller supplies nonnegative off-diagonal transition intensities in a square matrix. Diagonal values are ignored and replaced by the negative off-diagonal row sums. This permits reversible movement among nonterminal states and an absorbing terminal state when its off-diagonal row is zero.

Usage

```
rmtif_ctmc_generator(off_diagonal_rates)
```

Arguments

off_diagonal_rates	Square matrix of transition intensities.

Value

Valid CTMC generator with zero row sums.

Examples

```
rates <- matrix(c(0, 0.17, 0.29, 0), nrow = 2, byrow = TRUE)
rmtif_ctmc_generator(rates)
```

---

rmtif_gumbel_recovery_study

Summarize Gumbel regression recovery over simulated trials

Description

The summary contains the mean estimate, empirical SD, mean estimated SE, interval coverage, and convergence counts for every parameter and sample size. Trials with singular or nearly singular sensitivity are excluded from summary estimates and counted as failed fits.

Usage

```
rmtif_gumbel_recovery_study(n_per_arm_values,
    replications,
    beta_true,
    tau,
    sigma_x,
    conf_level,

    seed)
```

Arguments

n_per_arm_values	Per-arm sample sizes.
replications	Monte Carlo trials per sample size.
beta_true	True treatment and covariate coefficients.
tau	Restriction horizon.
sigma_x	Baseline covariate covariance matrix.
conf_level	Confidence level.
seed	Reproducibility seed.

Value

List containing replicate-level and summary data frames.

`rmtif_hfmock_analysis` *Analyze the public 'hfmock' scheduled-visit process*

Description

If 'data' is omitted, the function loads 'hfmock' only when the optional 'Wcompo' package is installed. No synthetic replacement is silently used. The analysis reports arm counts and terminal-aware IPCW bootstrap inference.

Usage

```
rmtif_hfmock_analysis(data = NULL,
    tau,
    visit_interval,
    bootstrap,
    seed,
    raw_status,

    ordered_state,
    quantile_probs,
    conf_level,
    bootstrap_success_fraction,

    minimum_survival = 0,
    projection_diagnostic = FALSE)
```

Arguments

`data`	Optional 'hfmock' data frame.
`tau`	Restriction horizon.
`visit_interval`	Scheduled visit spacing.
`bootstrap`	Stratified bootstrap replicates.
`seed`	Reproducibility seed.
`raw_status`	Named raw event-code vector; see 'rmtif_hfmock_subjects'.
`ordered_state`	Named ordered-state vector; see 'rmtif_hfmock_subjects'.
`quantile_probs`	Censor-time audit probabilities; see 'rmtif_hfmock_subjects'.
`conf_level`	Normal confidence level.
`bootstrap_success_fraction`	Required fraction of successful bootstrap resamples before reporting inference.
`minimum_survival`	Positivity threshold.
`projection_diagnostic`	Whether to retain the dense fixed-weight projection diagnostic.

Value

List with construction audit and an ‘rmtif_ipcw‘ fit.

`rmtif_hfmock_subjects` *Convert ‘Wcompo::hfmock‘ event records to scheduled-visit state paths*

Description

At caller-specified visits, the better nonterminal state means no hospitalization since the prior visit, the worse nonterminal state means at least one hospitalization, and death is an exact absorbing state. An end-of-record observation between visits censors the scheduled-visit process at the most recent completed visit. The raw data do not distinguish every such observation as loss to follow-up versus administrative closure. A record extending to or beyond the requested horizon supports carrying the most recent scheduled state to that horizon, including a horizon between scheduled visits.

Usage

```
rmtif_hfmock_subjects(data,
    tau,
    visit_interval,
    raw_status,
    ordered_state,
    quantile_probs)
```

Arguments

`data`	A data frame with ‘id‘, ‘time‘, ‘status‘, ‘Training‘, and ‘HF.etiology‘, such as ‘Wcompo::hfmock‘.
`tau`	Restriction horizon.
`visit_interval`	Scheduled-visit spacing.
`raw_status`	Named vector with ‘censor‘, ‘death‘, and ‘hospitalization‘ raw event codes.
`ordered_state`	Named vector with ‘no_hospitalization‘, ‘hospitalization‘, and ‘death‘ ordered-state codes.
`quantile_probs`	Three requested probabilities for the arm audit’s lower, median, and upper censor-time summaries.

Value

List with ‘rmtif_subject‘ objects and raw-data audit summaries.

`rmtif_ipcw` *Estimate terminal-aware arm-specific IPCW rmt-IF*

Description

The aim is unbiased marginal rmt-IF estimation under unconditional independent loss to follow-up within treatment arm. The estimator uses two arm-specific censoring Kaplan-Meier curves, carries observed death forward, and integrates over the union of state, death, and censoring times. Primary inference is a treatment-stratified bootstrap that refits both censoring curves in every replicate. The fixed-weight projection is returned only as a diagnostic because it omits censoring-curve estimation uncertainty.

Usage

```
rmtif_ipcw(subjects,
    tau,
    bootstrap,
    conf_level,
    bootstrap_success_fraction,

    seed = NULL,
    minimum_survival = 0,
    projection_diagnostic = TRUE,
    allow_insufficient_bootstrap = FALSE)
```

Arguments

`subjects`	List of observed ‘rmtif_subject‘ objects.
`tau`	Restriction horizon.
`bootstrap`	Number of stratified bootstrap replicates.
`conf_level`	Confidence level.
`bootstrap_success_fraction`	Minimum fraction of finite bootstrap estimates needed to report bootstrap inference.
`seed`	Optional reproducibility seed.
`minimum_survival`	Positivity threshold for each arm-specific curve.
`projection_diagnostic`	Whether to form the dense original-sample pair matrix for the fixed-weight projection diagnostic. Set to ‘FALSE‘ in a Monte Carlo study when only the IPCW point estimate and refitted-bootstrap inference are required.
`allow_insufficient_bootstrap`	If ‘TRUE‘, return a diagnostic object when too few bootstrap resamples retain support. Its bootstrap standard error and intervals are left undefined; this option is intended only to audit a boundary horizon, not to report primary inference.

Value

An ‘rmtif_fit‘ list with bootstrap inference when requested.

`rmtif_ipcw_equivalence_check` *Check the exact IPCW state-mass implementation against dense pair averaging*

Description

This audit helper evaluates one observed data set by both algebraically identical routes. It is useful for regression testing the faster bootstrap calculation; it does not fit a bootstrap or change the estimator.

Usage

```
rmtif_ipcw_equivalence_check(subjects,
    tau,
    minimum_survival = 0,
    tolerance = 1e-10)
```

Arguments

`subjects`	List of observed ‘rmtif_subject‘ objects.
`tau`	Restriction horizon.
`minimum_survival`	Positivity threshold for each arm-specific curve.
`tolerance`	Absolute numerical comparison tolerance.

Value

One-row data frame with dense, direct state-mass, and cached state-mass IPCW means, including a deterministic duplicate-subject bootstrap resample.

`rmtif_ipcw_simulation_parameters` *Prepare control and treatment CTMC generators for a censoring study*

Description

The design supplies every transition intensity and treatment multiplier. Additional trial design fields are passed through for convenience.

Usage

```
rmtif_ipcw_simulation_parameters(design)
```

Arguments

design	List containing 'rates_control', 'benefit_multiplier', 'initial_state', 'death_state', 'tau', 'primary_censoring_rates', 'allocations', 'trials', and 'bootstrap'.

Value

Named list of control, null-treatment and benefit-treatment CTMC generators, state labels, and supplied trial settings.

`rmtif_ipcw_simulation_study` *Run the dedicated differential-censoring IPCW study*

Description

The returned manuscript-ready summary contains truth, mean, bias, empirical SD, mean bootstrap SE, normal-interval coverage, rejection probability, and fit counts for every requested scenario/allocation combination.

Usage

```
rmtif_ipcw_simulation_study(allocations,
    scenarios,
    trials,
    bootstrap,
    tau,
    seed,
    minimum_survival,

    censoring_rates,
    parameters,
    conf_level,
    bootstrap_success_fraction,

    test_level,
    truth_grid_points)
```

Arguments

allocations	List of named 'c(n0, n1)' vectors.
scenarios	Null and/or treatment-benefit scenarios.
trials	Number of independently simulated trials.

bootstrap	Stratified bootstrap replicates per trial.
tau	Restriction horizon.
seed	Reproducibility seed.
minimum_survival	Positivity threshold.
censoring_rates	Named or ordered arm-specific exponential loss-to-follow-up rates used in every generated trial.
parameters	Markov generators and state labels.
conf_level	Confidence level.
bootstrap_success_fraction	Required successful bootstrap fraction.
test_level	Rejection threshold.
truth_grid_points	Number of numerical integration points for the population CTMC contrast.

Value

List containing replicate-level and summary data frames.

---

rmtif_monte_carlo_summary

Summarize Monte Carlo estimator and interval performance

Description

For each requested design cell, calculate the mean estimate, bias, empirical standard deviation, mean reported standard error, interval coverage of the supplied truth, two-sided zero-test rejection fraction, and successful-fit count. Additional trial-level columns can be averaged over all generated trials, as with observed loss-to-follow-up fractions.

Usage

```
rmtif_monte_carlo_summary(replicates,
    group_columns,
    estimate_column,
    se_column,
    truth_column,

    lower_column,
    upper_column,
    p_value_column,
    test_level,
    extra_mean_columns = character())
```

Arguments

`replicates`	Data frame with one row per simulated trial.
`group_columns`	Character names of the columns defining design cells.
`estimate_column`	Name of the numeric estimate column.
`se_column`	Name of the numeric estimated-standard-error column.
`truth_column`	Name of the numeric population-truth column.
`lower_column`	Name of the lower confidence-limit column.
`upper_column`	Name of the upper confidence-limit column.
`p_value_column`	Name of the two-sided zero-test p-value column.
`test_level`	Rejection threshold for the p-value.
`extra_mean_columns`	Optional trial-level numeric columns to average.

Value

One row per design cell with truth, mean estimate, bias, empirical SD, mean estimated SE, coverage, rejection, successful-fit count, total trial count, and means of requested extra columns.

`rmtif_observe_ctmc_path` *Apply loss to follow-up and terminal-event observation to a CTMC path*

Description

A terminal state reached before censoring is observed exactly and carried forward. Earlier loss to follow-up truncates the observed nonterminal path.

Usage

```
rmtif_observe_ctmc_path(path,
    arm,
    tau,
    censor_time,
    death_state,
    covariates = numeric(),

    id = NULL)
```

Arguments

path	Complete CTMC path from ‘rmtif_simulate_ctmc_path‘.
arm	Treatment indicator.
tau	Restriction horizon.
censor_time	Independent loss-to-follow-up time.
death_state	Absorbing terminal-state label.
covariates	Optional numeric baseline covariate vector.
id	Optional subject identifier.

Value

One observed ‘rmtif_subject‘.

---

rmtif_oncology_compare_trial

Compare six covariate-adjusted tests in one oncology trial

Description

OS/PFS use Cox Wald tests, OR and CDC use logistic Wald tests, study-specific ordinal BOR uses a proportional-odds Wald test, and RMT-IF tests its pairwise treatment coefficient. Every method uses the same generated subjects.

Usage

```
rmtif_oncology_compare_trial(simulation,
    rmt_inference,
    regression_bootstrap,
    conf_level,

    bootstrap_success_fraction,
    minimum_survival,
    test_level,
    bootstrap_seed = NULL)
```

Arguments

simulation	Output of ‘rmtif_simulate_oncology_trial()‘.
rmt_inference	Complete-path projection or censored IPCW bootstrap.
regression_bootstrap	IPCW bootstrap count (ignored for complete data).
conf_level	Confidence level.
bootstrap_success_fraction	Required successful bootstrap fraction.

`minimum_survival` IPCW positivity threshold.

`test_level` Two-sided rejection threshold.

`bootstrap_seed` Separate seed for the stratified bootstrap.

Value

Six-row long-format method result.

`rmtif_oncology_design` *Validate an ordered-state oncology simulation design*

Description

All baseline intensities, covariate and treatment log-rate effects, state labels, and the initial and terminal states are supplied by the caller. The named clinical states define the response-confirmation endpoints. No manuscript simulation settings are inserted by this constructor.

Usage

```
rmtif_oncology_design(transition,
    states,
    initial_state,
    death_state,
    scenario)
```

Arguments

`transition`	Data frame with ‘from‘, ‘to‘, ‘baseline_rate‘, ‘covariate_log_rate‘, and ‘treatment_log_rate‘ columns, one row per allowed transition. Rates are nonnegative; log-rate effects are finite.
`states`	Named, increasing, zero-based integer codes with names ‘CR‘, ‘PR‘, ‘SD‘, ‘PD‘, and ‘death‘, in clinical order from better to worse.
`initial_state`	Initial state code.
`death_state`	Absorbing terminal state code, equal to ‘states["death"]‘.
`scenario`	Nonempty scenario label used only in returned metadata.

Value

A validated list with ‘scenario‘, ‘transition‘, ‘states‘, ‘initial_state‘, and ‘death_state‘.

Examples

```
transition <- data.frame(
  from = c(2, 2, 1, 0, 3), to = c(1, 3, 0, 4, 4),
  baseline_rate = c(0.17, 0.23, 0.19, 0.13, 0.27),
  covariate_log_rate = rep(0, 5),
  treatment_log_rate = rep(0, 5))
design <- rmtif_oncology_design(transition,
  states = c(CR = 0L, PR = 1L, SD = 2L, PD = 3L, death = 4L),
  initial_state = 2L, death_state = 4L, scenario = "illustration")
design$transition
```

`rmtif_oncology_study` *Monte Carlo comparison of oncology endpoints on shared trials*

Description

Simulate each scenario using a separate seed for every trial and bootstrap. All six methods analyse the same subjects within a replicate. Failed fits remain in the replicate file; rejection proportions and their Monte Carlo standard errors use the number of valid fits as the denominator. Optional scenario checkpoints permit an interrupted run to resume without changing the random streams. Parallel workers are allocated by scenario, not method.

Usage

```
rmtif_oncology_study(scenarios,
    designs,
    n_per_arm,
    replications,
    tau,
    assessment_interval,

    assessment_jitter,
    censoring_rate,
    covariate_mean,
    covariate_sd,
    rmt_inference,

    regression_bootstrap,
    seed,
    conf_level,
    bootstrap_success_fraction,

    minimum_survival,
    test_level,
    workers = 1L,
    checkpoint_dir = NULL)
```

Arguments

scenarios	Distinct scenario labels to simulate.
designs	Named list of validated 'rmtif_oncology_design()' results, with one entry per requested scenario. List names and each design's 'scenario' label must agree.
n_per_arm	Number of subjects in each arm.
replications	Number of attempted trials per scenario.
tau	Administrative horizon, in months.
assessment_interval	Nominal interval between assessments.
assessment_jitter	Half-width of uniform assessment-time jitter.
censoring_rate	Exponential loss-to-follow-up rate; zero gives complete follow-up through the administrative horizon.
covariate_mean	Mean of the Gaussian baseline covariate.
covariate_sd	Positive standard deviation of the Gaussian covariate.
rmt_inference	Complete-path projection or censored IPCW bootstrap.
regression_bootstrap	Bootstrap resamples for censored RMT-IF inference.
seed	Master seed; replicate seeds are retained in the output.
conf_level	Confidence level for fitted models.
bootstrap_success_fraction	Required successful bootstrap fraction.
minimum_survival	IPCW positivity threshold.
test_level	Two-sided rejection threshold.
workers	Number of scenario-level PSOCK workers.
checkpoint_dir	Optional directory for resumable scenario checkpoints.

Value

List containing summary, replicate, transition-design, and runtime data frames. The example driver saves these outside the package.

---

rmtif_oncology_transition_rates

Calculate conditional oncology transition intensities and jump probabilities

---

Description

Apply the supplied Cox-form transition model at a specified treatment arm and baseline covariate value. The next-state probabilities divide each transition intensity by the total exit rate of its origin state. They are undefined ('NA') when all allowed exit rates from that state are zero.

Usage

```
rmtif_oncology_transition_rates(design,
    arm,
    covariate)
```

Arguments

design	Validated output of ‘rmtif_oncology_design()‘.
arm	Treatment-arm code, either control ‘0‘ or treatment ‘1‘.
covariate	Finite scalar baseline covariate value.

Value

The design’s transition data frame with ‘intensity‘, ‘total_exit_rate‘, and ‘next_transition_probability‘ columns added.

Examples

```
transition <- data.frame(
  from = c(2, 2, 1, 0, 3), to = c(1, 3, 0, 4, 4),
  baseline_rate = c(0.17, 0.23, 0.19, 0.13, 0.27),
  covariate_log_rate = rep(0, 5),
  treatment_log_rate = rep(0, 5))
design <- rmtif_oncology_design(transition,
  states = c(CR = 0L, PR = 1L, SD = 2L, PD = 3L, death = 4L),
  initial_state = 2L, death_state = 4L, scenario = "illustration")
rmtif_oncology_transition_rates(design, arm = 1, covariate = 0)
```

---

rmtif_pair_times *Compute pairwise restricted win, loss, and net times*

Description

This is the exact interval-by-interval calculation for two piecewise-constant paths. With ‘censoring = "none"‘, both paths must be complete through ‘tau‘. With ‘censoring = "arm_km"‘, the grid also contains every censoring time from ‘reference_subjects‘, and the integrand uses the terminal-aware, arm-specific inverse censoring weight described in the revised manuscript.

Usage

```
rmtif_pair_times(control,
    treatment,
    tau,
    censoring = c("none",
    "arm_km"),
    reference_subjects = NULL,

    minimum_survival = 0)
```

Arguments

`control`	One arm-0 ‘rmtif_subject‘.
`treatment`	One arm-1 ‘rmtif_subject‘.
`tau`	Restriction horizon.
`censoring`	Either ‘"none"‘ or ‘"arm_km"‘.
`reference_subjects`	Full sample used to estimate censoring curves.
`minimum_survival`	Positivity threshold for each censoring survival.

Value

Named vector with ‘win‘, ‘loss‘, and ‘net‘ in time units.

`rmtif_poisson_update_generator` *Convert a Poisson attempted-event update model to a CTMC generator*

Description

Attempted updates occur at the state-specific Poisson rates. Each attempt draws a destination from the corresponding row of ‘transition‘. A diagonal draw leaves the state unchanged; it is not a state transition. The resulting generator has off-diagonal entries ‘rates[i] * transition[i, j]‘ and rows summing to zero. This construction accommodates reversible movement and positive diagonal update probabilities.

Usage

```
rmtif_poisson_update_generator(transition,
    rates)
```

Arguments

`transition`	Row-stochastic attempted-update probability matrix.
`rates`	Positive attempted-update rates, one per state.

Value

CTMC generator matrix with the same row and column names.

Examples

```
transition <- matrix(c(0.37, 0.63, 0.44, 0.56),
                     nrow = 2, byrow = TRUE)
rmtif_poisson_update_generator(transition, rates = c(1.3, 2.1))
```

`rmtif_poisson_update_study` *Calibrate complete-path projection inference under Poisson updates*

Description

Repeatedly simulate two arms of a reversible ordered-state process and compute the complete-path rmt-IF curve at each requested horizon. The population truth follows from the corresponding CTMC generators. The returned summary reports bias, empirical SD, projection SE, coverage and a two-sided zero-contrast rejection proportion for every horizon.

Usage

```
rmtif_poisson_update_study(scenario,
    n_per_arm,
    replications,
    tau,
    increment,
    seed,
    design,

    conf_level,
    test_level,
    truth_grid_points)
```

Arguments

`scenario`	Either '"Null"' or an alternative design.
`n_per_arm`	Subjects per arm.
`replications`	Monte Carlo replications.
`tau`	Restricted horizon.
`increment`	Evaluation-grid spacing.
`seed`	Reproducibility seed.
`design`	List with 'transition_control', 'transition_treatment', 'rates_control', 'rates_treatment', 'states', 'initial_state', and 'absorbing_states'; all study values are supplied by the caller.
`conf_level`	Confidence level.
`test_level`	Rejection threshold.
`truth_grid_points`	Number of numerical integration points for the population CTMC contrast.

Value

List with replicate-level curves and a pointwise summary containing population truth, bias, empirical SD, mean projection SE, normal-interval coverage, and zero-null rejection proportion.

rmtif_projection_inference

Compute two-sample projection inference from pairwise net times

Description

The estimate is the mean of a control-by-treatment net-time matrix. Its standard error combines the empirical variance of row means divided by the control sample size with the variance of column means divided by the treatment sample size. A normal confidence interval and two-sided test of zero contrast use the supplied confidence level.

Usage

```
rmtif_projection_inference(pairwise_net_times,
    conf_level)
```

Arguments

pairwise_net_times	Finite numeric matrix with control subjects in rows and treatment subjects in columns.
conf_level	Confidence level for the normal interval.

Value

List with ‘estimate‘, ‘se‘, ‘conf_int‘, ‘statistic‘, ‘p_value‘, ‘row_projection‘, and ‘column_projection‘.

Examples

```
net <- matrix(c(1, 2, 3, 4), nrow = 2, byrow = TRUE)
rmtif_projection_inference(net, conf_level = 0.95)
```

rmtif_regression

Fit covariate-adjusted pairwise rmt-IF regression

Description

The model uses ‘treatment + (X_treatment - X_control)‘ as its pairwise design, so the first coefficient is the treatment effect for equal baseline covariates. The score is exactly the manuscript score ‘(win - loss * exp(beta^T difference)) * difference‘. With arm-specific IPCW, a stratified bootstrap refits both censoring curves and is the primary inference; the fixed-weight sandwich is retained as a diagnostic. A singular or nearly singular score-sensitivity matrix makes coefficients and intervals unidentified: the function raises an error instead of reporting a pseudoinverse-based, spuriously narrow interval. Near singular means the smallest singular value is at most the largest times square-root machine precision; remove redundant covariates and refit.

Usage

```
rmtif_regression(subjects,
    tau,
    censoring = c("none",
    "arm_km"),
    bootstrap,
    conf_level,

    bootstrap_success_fraction,
    seed = NULL,
    minimum_survival = 0,
    beta_initial = NULL)
```

Arguments

subjects	List of ‘rmtif_subject‘ objects.
tau	Restriction horizon.
censoring	Either ‘"none"‘ or terminal-aware ‘"arm_km"‘.
bootstrap	Stratified bootstrap replicates. ‘0‘ returns fixed-weight diagnostic inference only.
conf_level	Confidence level.
bootstrap_success_fraction	Minimum fraction of finite bootstrap fits needed to report bootstrap inference.
seed	Optional reproducibility seed.
minimum_survival	Positivity threshold for each censoring curve.
beta_initial	Optional starting coefficient vector.

Value

An ‘rmtif_regression‘ fit with treatment and covariate coefficients.

---

rmtif_simulate_ctmc_path

Simulate a complete continuous-time Markov state path

Description

Wait at each state according to its generator exit intensity, then choose the next state in proportion to the off-diagonal intensities. An absorbing state ends transitions while retaining its state through the horizon.

Usage

```
rmtif_simulate_ctmc_path(generator,
    initial_state,
    tau)
```

Arguments

generator	Valid CTMC generator matrix.
initial_state	Initial state index starting at zero.
tau	Follow-up horizon.

Value

List with right-continuous ‘times‘ and ‘states‘ vectors.

Examples

```
rates <- matrix(c(0, 0.17, 0.29, 0), nrow = 2, byrow = TRUE)
generator <- rmtif_ctmc_generator(rates)
set.seed(397)
rmtif_simulate_ctmc_path(generator, initial_state = 0, tau = 3.7)
```

---

rmtif_simulate_gumbel_regression

Simulate the constant-outcome Gumbel regression DGM

Description

This descriptive entry point generates the same Gaussian-covariate, Gumbel-error pairwise-regression trial as ‘rmtif_simulate_gumbel_trial‘. All treatment, slope, covariance, horizon, and sample-size inputs come from the caller.

Usage

```
rmtif_simulate_gumbel_regression(n_per_arm,
    beta_true,
    tau,
    sigma_x,
    seed = NULL)
```

Arguments

n_per_arm	Subjects per treatment arm.
beta_true	Named treatment-intercept and covariate-slope vector.
tau	Restricted time horizon.
sigma_x	Positive-definite baseline-covariate covariance matrix.
seed	Optional reproducibility seed.

Value

List with complete ordered-state subjects, true parameter vector, and restriction horizon.

`rmtif_simulate_gumbel_trial` *Simulate a Gumbel pairwise-regression data-generating mechanism*

Description

Generate Gaussian baseline covariates with the supplied covariance matrix, treatment allocation, and independent standard Gumbel errors. The resulting continuous latent utility is constant over follow-up and is represented by its negative as an ordered state, so smaller states mean better outcomes. This is a regression-equation sanity check rather than a multistate DGM.

Usage

```
rmtif_simulate_gumbel_trial(n_per_arm,
    beta_true,
    tau,
    sigma_x,
    seed = NULL)
```

Arguments

`n_per_arm`	Subjects per arm.
`beta_true`	Named vector: treatment intercept followed by slopes.
`tau`	Restriction horizon.
`sigma_x`	Positive-definite covariance matrix for baseline covariates.
`seed`	Optional seed.

Value

Complete ‘rmtif_subject‘ list and true parameter.

Examples

```
trial <- rmtif_simulate_gumbel_trial(
  n_per_arm = 12, beta_true = c(0.2, 0.3), tau = 3.7,
  sigma_x = matrix(1), seed = 397)
rmtif_complete(trial$subjects, tau = 3.7, conf_level = 0.95)
```

`rmtif_simulate_ipcw_trial` *Simulate one dedicated IPCW trial*

Description

The aim is to evaluate the marginal arm-specific IPCW estimator under a reversible nonterminal process, absorbing death, unequal allocation, and differential but within-arm independent loss to follow-up.

Usage

```
rmtif_simulate_ipcw_trial(n0,
    n1,
    scenario,
    tau,
    censoring_rates,
    parameters,
    seed = NULL)
```

Arguments

`n0`	Control sample size.
`n1`	Treatment sample size.
`scenario`	‘"Null"‘ or ‘"Treatment benefit"‘.
`tau`	Restriction horizon.
`censoring_rates`	Arm-specific exponential loss-to-follow-up rates.
`parameters`	Markov generators and state labels.
`seed`	Optional seed.

Value

List of observed subjects and the two population generators.

`rmtif_simulate_oncology_trial` *Simulate one shared Appendix C oncology trial*

Description

All six methods are derived from this same patient-level realization. Nominal visits are generated far enough to cover the jitter window and then visits outside ‘(0,tau]‘, after loss to follow-up, or at/after death are dropped. The RMT-IF representation is compressed to actual observed state changes; the complete visit history is retained for OR and BOR confirmation.

Usage

```
rmtif_simulate_oncology_trial(n_per_arm,
    design,
    tau,
    assessment_interval,
    assessment_jitter,

    censoring_rate,
    covariate_mean,
    covariate_sd,
    seed = NULL)
```

Arguments

n_per_arm	Exact subjects in each randomized arm.
design	Validated output of 'rmtif_oncology_design()'.
tau	Administrative restriction horizon.
assessment_interval	Nominal imaging interval.
assessment_jitter	Half-width of independent uniform visit jitter.
censoring_rate	Exponential LTFU rate; zero specifies complete follow-up.
covariate_mean	Mean of the Gaussian baseline covariate.
covariate_sd	Positive standard deviation of the Gaussian covariate.
seed	Reproducibility seed.

Value

Shared endpoint data, RMT-IF subjects, visits, and latent paths.

Examples

```
transition <- data.frame(
  from = c(2, 2, 1, 0, 3), to = c(1, 3, 0, 4, 4),
  baseline_rate = c(0.17, 0.23, 0.19, 0.13, 0.27),
  covariate_log_rate = rep(0, 5),
  treatment_log_rate = rep(0, 5))
design <- rmtif_oncology_design(transition,
  states = c(CR = 0L, PR = 1L, SD = 2L, PD = 3L, death = 4L),
  initial_state = 2L, death_state = 4L, scenario = "illustration")
trial <- rmtif_simulate_oncology_trial(
  n_per_arm = 8, design = design, tau = 3.7,
  assessment_interval = 0.8, assessment_jitter = 0.07,
  censoring_rate = 0, covariate_mean = 0, covariate_sd = 1,
  seed = 397)
names(trial)
```

`rmtif_simulate_poisson_updates` *Simulate finite-state Poisson attempted-event update trajectories*

Description

At each state an exponential waiting time with the specified attempted-event rate precedes a destination draw from ‘transition‘. A self-update advances time while retaining the same state. The output is one complete observed ordered-state trajectory per subject; the supplied state order determines favorable versus unfavorable states in later rmt-IF calculations.

Usage

```
rmtif_simulate_poisson_updates(n,
    transition,
    rates,
    tau,
    states,
    initial_state,
    arm,
    absorbing_states,

    seed = NULL)
```

Arguments

`n`	Number of subjects.
`transition`	Row-stochastic attempted-update probability matrix.
`rates`	Positive attempted-update rates, one per state.
`tau`	Follow-up horizon.
`states`	Ordered numeric state labels.
`initial_state`	Initial state label.
`arm`	Treatment arm, 0 or 1.
`absorbing_states`	Optional explicit absorbing state labels.
`seed`	Optional seed.

Value

List of complete ‘rmtif_subject‘ trajectories.

Examples

```
transition <- matrix(c(0.37, 0.63, 0.44, 0.56),
                     nrow = 2, byrow = TRUE)
paths <- rmtif_simulate_poisson_updates(
  n = 4, transition = transition, rates = c(1.3, 2.1),
  tau = 3.7, states = c(0, 1), initial_state = 0,
  arm = 0, absorbing_states = numeric(), seed = 397)
length(paths)
```

`rmtif_state_at` *Evaluate a subject's right-continuous state*

Description

This helper is intended for transparent checking of the path construction. It returns 'NA' after unobserved loss to follow-up, but continues to return the absorbing terminal state after an observed death.

Usage

```
rmtif_state_at(subject,
    time)
```

Arguments

`subject`	An 'rmtif_subject'.
`time`	One or more nonnegative evaluation times.

Value

A numeric vector of states, with 'NA' where the path is unavailable.

`rmtif_subject` *Construct one observed ordered-state trajectory*

Description

The function creates the common subject representation used by every rmt-IF estimator in this file. 'times' and 'states' define a right-continuous, piecewise-constant observed path. An observed terminal event is inserted at 'death_time' and carried forward; it never truncates the outcome path. 'censor_time' is loss to follow-up or administrative end, and 'censor_event' distinguishes loss to follow-up from administrative end.

Usage

```
rmtif_subject(arm,
    times,
    states,
    tau,
    covariates = numeric(),
    censor_time = tau,

    censor_event = FALSE,
    death_time = Inf,
    death_state = NULL,
    id = NULL)
```

Arguments

arm	Treatment indicator, 0 for control or 1 for treatment.
times	Numeric state-observation/change times and must include baseline.
states	Ordered states corresponding to ‘times‘; smaller is better.
tau	Planned restriction horizon.
covariates	Optional numeric baseline covariate vector.
censor_time	Loss-to-follow-up or administrative time.
censor_event	Whether ‘censor_time‘ is a loss-to-follow-up event.
death_time	Observed terminal-event time, or ‘Inf‘ if unobserved/absent.
death_state	Ordered absorbing terminal state, or ‘NULL‘ if not used.
id	Optional subject identifier.

Value

An object of class ‘rmtif_subject‘.

Examples

```
path <- rmtif_subject(arm = 1, times = c(0, 1.5),
                      states = c(2, 1), tau = 3.7)
rmtif_state_at(path, c(0, 1.5, 3))
```

---

rmtif_true_ctmc_contrast

Compute a population CTMC rmt-IF contrast

Description

The aim is to obtain simulation truth without Monte Carlo error. Transition probabilities are computed by uniformization and the pairwise state contrast is integrated numerically.

Usage

```
rmtif_true_ctmc_contrast(generator_control,
    generator_treatment,
    initial_state,
    tau,
    grid_points)
```

Arguments

generator_control	Control CTMC generator.
generator_treatment	Treatment CTMC generator.
initial_state	Common initial state, numbered from zero.
tau	Restriction horizon.
grid_points	Number of equally spaced integration points, supplied by the caller to control numerical accuracy.

Value

Population treatment-minus-control rmt-IF in time units.

Examples

```
rates <- matrix(c(0, 0.17, 0.29, 0), nrow = 2, byrow = TRUE)
generator <- rmtif_ctmc_generator(rates)
rmtif_true_ctmc_contrast(generator, generator, initial_state = 0,
                         tau = 3.7, grid_points = 51)
```

Index

RMTIF *(*RMTIF-package*)*, 2
RMTIF-package, 2
rmtif_censoring_support, 3
rmtif_complete, *3*, 4
rmtif_complete_curve, 4
rmtif_complete_state_count_curve, 5
rmtif_construct_scheduled_visit_states, 6
rmtif_ctmc_generator, 6
rmtif_gumbel_recovery_study, 7
rmtif_hfmock_analysis, 8
rmtif_hfmock_subjects, 9
rmtif_ipcw, *3*, 10
rmtif_ipcw_equivalence_check, 11
rmtif_ipcw_simulation_parameters, 11
rmtif_ipcw_simulation_study, 12
rmtif_monte_carlo_summary, 13
rmtif_observe_ctmc_path, 14
rmtif_oncology_compare_trial, 15
rmtif_oncology_design, 16
rmtif_oncology_parameters *(*rmtif_oncology_design*)*, 16
rmtif_oncology_study, 17
rmtif_oncology_transition_rates, 18
rmtif_pair_times, 19
rmtif_poisson_update_generator, 20
rmtif_poisson_update_study, 21
rmtif_projection_inference, 22
rmtif_regression, *3*, 22
rmtif_simulate_ctmc_path, 23
rmtif_simulate_gumbel_regression, 24
rmtif_simulate_gumbel_trial, 25
rmtif_simulate_ipcw_trial, 26
rmtif_simulate_oncology_trial, 26
rmtif_simulate_poisson_updates, 28
rmtif_state_at, 29
rmtif_subject, *3*, 29
rmtif_true_ctmc_contrast, 30