Package 'multiswc'

May 29, 2026

Type Package

Title Multi-Regime Marginal Structural Cox Model for Multi-Way Treatment Switching in Oncology Clinical Trials with Survival Endpoints

Version 0.1.2

Description Estimate the causal effect of sustained treatment strategies on overall survival in clinical trials with possible treatment crossover and switch to subsequent therapy. Simulate faithful longitudinal clinical trials data with survival endpoints and multi-way treatment switches allowing for time-dependent prognostic factors. For more on methodological background, please see: Keogh and colleagues (2021) <doi:10.1002/bimj.202000040> and Suarez and colleagues (2008) <doi:10.1016/j.jclinepi.2007.11.007>.

License MIT + file LICENSE

Imports nnet, survival

Encoding UTF-8

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0), knitr, rmarkdown

URL https://github.com/tonyhbc/multiswc

BugReports https://github.com/tonyhbc/multiswc/issues

NeedsCompilation no

Author Haobin Chen [aut, cre],
Yuxuan Chen [aut],
Philip He [aut]

Maintainer Haobin Chen <tony.haobin.chen@alumni.emory.edu>

Repository CRAN

Date/Publication 2026-05-29 10:00:02 UTC

Contents

`multimsm` *Fit a Multi-regime Marginal Structural Cox Model for Three-way Treatment Switching*

Description

`multimsm()` fits a marginal structural Cox model with categorical treatment for randomized clinical trials or trial-like longitudinal data with multi-way treatment switching, assuming participants underwent regular longitudinal follow-up visits. This method assumes a three-way treatment switch scenario: control participants may crossover to experimental group and both control and experimental participants may switch to a subsequent therapy. `multimsm()` takes a longitudinal survival dataset with 3 treatment process variables as primary inputs:

- `rand`: **baseline** randomized treatment arm. `1` denotes randomized experimental treatment and `0` denotes randomized control/SOC.
- `cross`: **time-varying** control-to-experimental crossover indicator (1/0).
- `subseq`: **time-varying** subsequent therapy initiation indicator (1/0).

Internally, `multimsm()` converts this triplet of treatment trajectory into the five-level *time-varying treatment regime*:

$$G(t) \in \{C, E, CE, CS, ES\}$$

where `C` and `E` denote **sustained control** and **sustained experimental** treatment regime at t, `CE` denotes **control-to-experimental crossover**, `CS` denotes on **control-to-subsequent switch** regime status at t, and `ES` denotes on **experimental-to-subsequent switch** regime status at t.

The downstream estimator is a multinomial-regime stabilized-weight marginal structural Cox model with the time-varying regime as predictor:

$$\lambda_{T^{\overline{g}}}(t) = \lambda_0(t) \exp\{\beta_e \mathbb{I}(G(t) = E) + \beta_{ce} \mathbb{I}(G(t) = CE) + \beta_{cs} \mathbb{I}(G(t) = CS) + \beta_{es} \mathbb{I}(G(t) = ES)\}$$

The stabilized treatment-regime weight (S-IPTW) at each post-baseline interval is the probability of the observed current regime under a stabilizer model divided by the corresponding probability under a denominator model. The cumulative product of these row-specific ratios is used as the inverse probability weight. Optional ordinary censoring weights may also be multiplied in, if informative censoring is present.

Usage

```
multimsm(dat_long, id = "id", tstart = "t.start", tstop = "t.stop",
  event = "event", cens = NULL, rand = "rand", cross = "cross",
  subseq = "subseq", base_cov = NULL, iptw_num = NULL, iptw_den,
  ipcw_mod = NULL, wt_trunc = NULL, prob_bounds = c(1e-06, 1 - 1e-06),
  normalize_weights = TRUE, robust = TRUE, check_inputs = TRUE,
  trace_multinom = FALSE, maxit = 200)
```

Arguments

dat_long	A data.frame in long-format start-stop format with one row per subject follow-up interval.
id	Character. Subject identifier variable name.
tstart	Character. Interval start-time variable name.
tstop	Character. Interval stop-time variable name.
event	Character. Event indicator variable name (1: event, 0: censor).
cens	Optional character Ordinary right-censoring indicator used only if ipcw_mod is supplied. This is not the treatment-switching process.
rand	Character. Baseline randomization variable name (1: randomized experimental treatment, 0: randomized control). The value must be constant within subject over time.
cross	Character. *Time-varying* control-to-experimental crossover indicator. It may be one-time pulse at crossover time (...,0,1,0,0,...) or absorbing (...,0,1,1,1,...). It must be structurally 0 always for subjects randomized to experimental treatment (assumed no EC switch).
subseq	Character. *Time-varying* subsequent therapy initiation indicator. It may be pulse or absorbing.
base_cov	Optional character vector of baseline covariates to include in the default numerator model when iptw_num is NULL.
iptw_num	Optional *right-hand-side* formula or character string for the stabilized numerator multinomial regime model. If NULL, the default is ~ regime_lag + factor(visit) + < base_cov >.
iptw_den	Required *right-hand-side formula* or character string for the denominator multinomial regime model. This should typically include regime_lag, visit or time terms, baseline covariates, and time-varying confounders affected by prior treatment and predictive of subsequent switching.
ipcw_mod	Optional right-hand-side formula or character string for an ordinary censoring model. If supplied, cens must also be supplied.
wt_trunc	Optional numeric in (0.5, 1). If supplied, the final combined inverse probability weight is truncated at the (1 - wt_trunc) and wt_trunc empirical quantiles before the Cox model is fit. For example, wt_trunc = 0.95 caps the lower and upper tails at the 5th and 95th percentiles. The default NULL performs no quantile truncation.
prob_bounds	Numeric length-2 vector of lower and upper probability bounds used to stabilize extreme predicted probabilities away from 0 and 1.

`normalize_weights`
: Logical. If `TRUE`, normalize the final truncated weights to have mean 1.

`robust`
: Logical. If `TRUE`, fit the Cox model with `cluster(id)` to use the standard robust sandwich variance estimator.

`check_inputs`
: Logical. If `TRUE`, enforce structural checks for the five-regime treatment-switching system.

`trace_multinom`
: Logical. Passed to `nnet::multinom()`.

`maxit`
: Integer maximum number of iterations for the multinomial models.

Details

The treatment-switching process is assumed to follow the package's five-regime structure G(t):

$$G_i(t) = \begin{cases} C, & R_i = 0,\ Crs_i(t) = 0,\ Subs_i(t) = 0, \\ E, & R_i = 1,\ Crs_i(t) = 0,\ Subs_i(t) = 0, \\ CE, & R_i = 0,\ Crs_i(t) = 1,\ Subs_i(t) = 0, \\ CS, & R_i = 0,\ Crs_i(t) = 0,\ Subs_i(t) = 1, \\ ES, & R_i = 1,\ Crs_i(t) = 0,\ Subs_i(t) = 1. \end{cases}$$

`R(t)` is `rand` randomized arm, `Crs(t)` is `cross` the absorbing crossover status, and `Subs(t)` is `subseq` the absorbing subsequent initiation status. Users may supply `cross` and `subseq` with either sustained absorbing status indicator such as (`0, 0, 1, 1, 1`), or as switch-initiation action indicators such as (`0, 0, 1, 0, 0`) - both acceptable.

The current method allows *only one switch type* per subject's follow up. Thus, control-arm subjects may remain in `C`, transition to `CE`, or transition to `CS`; experimental-arm subjects may remain in `E` or transition to `ES`.

The outcome model uses generated `regime` as the exposure. By default `C` **is the reference regime**, so the coefficient for `regimeE` estimates the marginal sustained experimental-versus-control contrast aimed by the package's primary no-switch estimand. The remaining coefficients describe the distinct switched-regime contrasts averaging switch times relative to sustained control.

Value

An object of class `"multimsm"`; a list with components:

- `coef_table` A `data.frame` of full estimated coefficient table.
- `coef` Log-hazard ratio estimates for treatment regimes against control.
- `hr` Hazard ratio estimates for treatment regimes against control.
- `hr_ci` Wald 95% confidence intervals for the hazard ratios.
- `fit` The fitted weighted Cox model from `survival::coxph()`.
- `p.val` P values of regime effect estimates.
- `dat_long` Augmented long-format data containing the derived variables, including switch indicators, and final estimated weights (`.w_use`/`.w_use_trunc`).
- `regime_models` A list containing the fitted numerator and denominator mulinomial regime models.

- `censor_model` The fitted censoring model when `ipcw_mod` is supplied; othrwise `NULL`.
- `diagnostics` A list containing untruncated final-weight quantiles, final switch-regime counts/proportions, cohort size, and truncation bounds.
- `call` The matched function call.

References

Robins JM, Hernan MA, Brumback B. Marginal structural models and causal inference in epidemiology. *Epidemiology*. 2000;11(5):550-560.

Suarez D, Haro JM, Novick D, et al. Marginal structural models for multiple treatment comparisons: an application to antipsychotic treatment for schizophrenia. *Journal of Clinical Epidemiology*. 2008;61(6):525-530.

Examples

```
set.seed(123)

sim_obj <- simswc(
  n = 300,
  n_visit = 8,
  base_cov = c("L1", "L3"),
  trt_prob = 0.5,
  param_tdcov = c(-1.5, 0.3, 0.3, -0.2, 0.5),
  param_tdcov2 = c(0.05, 0.2, 0.2, -0.2, 0.7),
  param_sw = c(-3, 0.4, 0.4, -0.3, 0.3, 0.8),
  param_haz = c(0.1, log(0.8), log(1.1), log(1.1),
                log(1.4), log(1.4), log(1.3), log(1), log(0.9)),
  param_cens = NULL,
  param_select = c(0, 0.5, 0.25),
  lagk = TRUE,
  true_hr = FALSE
)

# Convert to the required treatment switch indicator triplet
dat <- sim_obj$dat_long
dat$rand <- stats::ave(dat$A, dat$id, FUN = function(x) rep(x[1], length(x)))
dat$cross <- dat$S_CE
dat$subseq <- pmax(dat$S_ES, dat$S_CS)

fit <- multimsm(
  dat_long = dat, id = "id", tstart = "t.start", tstop = "t.stop",
  event = "event", rand = "rand", cross = "cross", subseq = "subseq",
  base_cov = c("L1", "L3"),
  iptw_num = ~ regime_lag + factor(visit) + L1 + L3,
  iptw_den = ~ regime_lag + factor(visit) + L1 + L3 + X + U + Alag1,
  wt_trunc = 0.95
)

fit
```

print.multimsm *Print a fitted multi-regime marginal structural Cox model*

Description

Print a fitted multi-regime marginal structural Cox model

Usage

```
## S3 method for class 'multimsm'
print(x, digits = 3, ...)
```

Arguments

x	An object returned by multimsm().
digits	Number of decimal places for coefficient and weight summaries.
...	Currently unused.

Value

Invisibly returns x.

simswc *Simulate longitudinal survival data with three-way treatment switching*

Description

simswc() simulates a two-arm randomized clinical-trial-like longitudinal survival dataset with a three-way post-randomization treatment swich process. It is intended to generate data for evaluating and demonstrating multi-regime marginal structural Cox model multimsm().

Follow-up is represented on a discrete visit grid $k = 0, 1, \ldots, K-1$, where K = n_visit. Each subject is randomized at baseline to experimental treatment $A_0 = 1$ or control treatment $A_0 = 0$. During follow-up, subjects may experience at most one absorbing treatment switch:

- ES: randomized experimental $\rightarrow$ subsequent therapy;
- CE: randomized control $\rightarrow$ experimental crossover;
- CS: randomized control $\rightarrow$ subsequent therapy.

The simulator generates baseline covariates, two time-varying confounders (X and U), a generic switch process V, switch-type indicators S_ES, S_CE, and S_CS, optional random censoring, and a survival outcome from a piecewise exponential hazard model. The output includes both subject-level wide-format data and counting-process long-format data suitable for Cox modeling and for the package's multi-regime MSM workflow.

Usage

```
simswc(n, n_visit, base_cov = paste0("L", 1:6), trt_prob = 0.5,
  param_tdcov, param_tdcov2, param_sw, param_haz, param_cens = NULL,
  study_end = n_visit, lagk = FALSE, param_select = NULL,
  p_pick_CE = 0.5, true_hr = TRUE, truth_n = 1e+05)
```

Arguments

n	Integer sample size. The current implementation requires an integer greater than 49.
n_visit	Integer number of planned discrete visits, K. The visit indices in the wide data are `0, ..., n_visit - 1`, and the long-format intervals are initially `[0,1), [1,2), ...`.
base_cov	Character vector specifying which baseline covariates are included in the simulator's linear predictors and used to define strata for baseline randomization. Must be a non-empty subset of `c("L1", "L2", "L3", "L4", "L5", "L6")`. Coefficients in all parameter vectors must follow this exact order.
trt_prob	Numeric scalar in `(0, 1)`. Target probability of assignment to baseline experimental treatment $A_0 = 1$ within each stratum formed by `base_cov`.
param_tdcov	Numeric vector of length `length(base_cov) + 3` giving the coefficients η for the binary time-varying confounder model for `X`. If `p = length(base_cov)`, the required order is: • `[1]`: intercept η_0; • `[2:(p+1)]`: baseline-covariate coefficients η_L, in the same order as `base_cov`; • `[p+2]`: coefficient for prior treatment A_{k-1} η_A; • `[p+3]`: coefficient for prior binary confounder X_{k-1}, η_X.
param_tdcov2	Numeric vector of length `length(base_cov) + 3` giving the coefficients α for the continuous time-varying confounder mean model for `U`. If `p = length(base_cov)`, the required order is: • `[1]`: intercept α_0; • `[2:(p+1)]`: baseline-covariate coefficients α_L, in the same order as `base_cov`; • `[p+2]`: coefficient for prior treatment A_{k-1}, α_A; • `[p+3]`: coefficient for prior continuous confounder U_{k-1}, α_U.
param_sw	Numeric vector of length `length(base_cov) + 4` giving the coefficients δ for the generic treatment-switch model. If `p = length(base_cov)`, the required order is: • `[1]`: intercept δ_0; • `[2:(p+1)]`: baseline-covariate coefficients δ_L, in the same order as `base_cov`; • `[p+2]`: coefficient for prior treatment A_{k-1}, δ_A; • `[p+3]`: coefficient for current binary confounder X_k, δ_X; • `[p+4]`: coefficient for current continuous confounder U_k, δ_U.
param_haz	Numeric vector of length `length(base_cov) + 7` specifying the piecewise exponential survival model. If `p = length(base_cov)`, the required order is:

- [1]: baseline hazard scale λ_0. This is a hazard scale, not a log-hazard coefficient, and should be positive;
- [2]: log hazard ratio for current treatment A_k, β_A;
- [3:(p+2)]: baseline-covariate log hazard ratios β_L, in the same order as base_cov;
- [p+3]: log hazard ratio for current binary confounder X_k, β_X;
- [p+4]: log hazard ratio for current continuous confounder U_k, β_U;
- [p+5]: log hazard ratio for ES status, β_{ES};
- [p+6]: log hazard ratio for CE status, β_{CE};
- [p+7]: log hazard ratio for CS status, β_{CS}.

param_cens — Optional numeric vector of length length(base_cov) + 4 giving the coefficients γ for the discrete-time random censoring model. If NULL, no random censoring is generated beyond the administrative censoring mechanism. If p = length(base_cov), the required order is:

- [1]: intercept γ_0;
- [2:(p+1)]: baseline-covariate coefficients γ_L, in the same order as base_cov;
- [p+2]: coefficient for current treatment A_k, γ_A;
- [p+3]: coefficient for current binary confounder X_k, γ_X;
- [p+4]: coefficient for current continuous confounder U_k, γ_U.

study_end — Numeric scalar giving the nominal administrative study-end time. Must be between 2 and n_visit. The actual administrative censoring time returned as C.a is study_end - T_e, where T_e is a randomly sampled integer from 0:floor(n_visit / 3).

lagk — Logical. If TRUE, add lagged versions of A, X, and U up to three prior visits and one-step future versions of X and U to the long-format output. If FALSE, these auxiliary lag/lead variables are not added.

param_select — Optional numeric vector of length 3 controlling the CE versus CS destination model among newly switching control-arm subjects. The required order is:

- [1]: intercept ξ_0;
- [2]: coefficient for current X_k, ξ_X;
- [3]: coefficient for current U_k, ξ_U.

If NULL, p_pick_CE is used instead.

p_pick_CE — Numeric scalar in [0, 1]. Probability that a newly switching control-arm subject switches to experimental therapy (CE) rather than subsequent therapy (CS) when param_select = NULL.

true_hr — Logical. If TRUE, compute an approximate Monte Carlo benchmark for the sustained always-experimental versus always-control comparison using truth_n simulated subjects per sustained regime. This benchmark ignores switching and uses the non-switch components of the longitudinal confounder and hazard models.

truth_n — Non-negative integer Monte Carlo sample size used only when true_hr = TRUE. Set true_hr = FALSE or truth_n = 0 to skip this potentially time-consuming benchmark.

Details

Let L_i denote the vector of selected baseline covariates named in `base_cov`. The simulator first generates six possible baseline covariates:

- `L1`: Binary age group indicator, $L_1 \sim \text{Bernoulli}(0.3)$.
- `L2`: Three-level risk group taking values `0`, `1`, and `2` with probabilities `0.60`, `0.35`, and `0.05`, respectively.
- `L3`: Binary baseline metastasis/progression-like indicator, $L_3 \sim \text{Bernoulli}(0.1)$.
- `L4`: Continuous tumor-size-like marker, $L_4 \sim N(60, 35^2)$.
- `L5`: Binary biomarker indicator, $L_5 \sim \text{Bernoulli}(0.8)$.
- `L6`: Binary prior-treatment indicator, $L_6 \sim \text{Bernoulli}(0.6)$.

Only the variables included in `base_cov` are used in the treatment, confounder, switch, censoring, and hazard linear predictors, and only these selected baseline covariates are returned in the simulated datasets.

Baseline treatment is assigned by stratified randomization within the interaction strata formed by `base_cov`; within each stratum, approximately `trt_prob` of subjects are assigned to $A_0 = 1$. Because `base_cov` defines both the model covariate vector and the randomization strata, using a continuous variable such as `L4` may create many small strata. In practical simulation studies, it is often preferable to stratify on categorical or binary baseline covariates, for example `base_cov = c("L1", "L3")`.

The time-varying binary confounder `X` is initialized as $X_0 = 0$ and is made absorbing by construction. For $k = 1, \ldots, K-1$,

$$\Pr(\widetilde{X}_{ik} = 1 \mid H_{i,k-1}) = \text{expit}\{\eta_0 + \eta_L^\top L_i + \eta_A A_{i,k-1} + \eta_X X_{i,k-1}\},$$

and the stored value is

$$X_{ik} = \max(X_{i,k-1}, \widetilde{X}_{ik}).$$

The continuous time-varying confounder `U` is initialized as $U_0 \sim N(0,1)$. For $k = 1, \ldots, K-1$,

$$U_{ik} \sim N(\mu_{ik}, 1), \quad \mu_{ik} = \alpha_0 + \alpha_L^\top L_i + \alpha_A A_{i,k-1} + \alpha_U U_{i,k-1}.$$

The generic switch indicator `V` is initialized as $V_0 = 0$. Among subjects who have not yet switched, the probability of a new switch at visit k is

$$\Pr(V_{ik} = 1 \mid V_{i,k-1} = 0, H_{ik}) = \text{expit}\{\delta_0 + \delta_L^\top L_i + \delta_A A_{i,k-1} + \delta_X X_{ik} + \delta_U U_{ik}\}.$$

Once a subject switches, `V` remains equal to 1. The switch destination is then assigned as follows:

- if $A_{i0} = 1$, the subject switches to `ES`; `S_ES` is set to 1 from that visit onward and `A` is set to 0 thereafter to represent being off the original experimental-treatment indicator and on subsequent therapy;
- if $A_{i0} = 0$, the subject switches either to `CE` or to `CS`; under `CE`, `S_CE` is set to 1 and `A` is set to 1 thereafter; under `CS`, `S_CS` is set to 1 and `A` remains 0 thereafter.

If `param_select = NULL`, the control-arm switch destination is chosen with fixed probability `p_pick_CE` for `CE` and `1 - p_pick_CE` for `CS`. If `param_select` is supplied, the probability of `CE` among newly switching control-arm subjects is

$$\Pr(\mathrm{CE}_{ik} = 1 \mid \text{new switch}, A_{i0} = 0) = \mathrm{expit}\{\xi_0 + \xi_X X_{ik} + \xi_U U_{ik}\}.$$

The event time is generated from a piecewise exponential model. For interval $[k, k+1)$, where $k = 0, \ldots, K-1$, the internal column index is $m = k + 1$, and the interval-specific hazard is

$$\lambda_{ik} = \{\lambda_0 \exp(0.1m)\} \exp\{\beta_A A_{ik} + \beta_L^\top L_i + \beta_X X_{ik} + \beta_U U_{ik} + \beta_{ES} S_{ik}^{ES} + \beta_{CE} S_{ik}^{CE} + \beta_{CS} S_{ik}^{CS}\}.$$

Conditional on the interval covariates, an exponential waiting time with rate λ_{ik} is drawn. If the waiting time is less than 1, the event is placed inside that interval at exact continuous time $k + T^*$; otherwise simulation proceeds to the next interval.

If `param_cens` is supplied, random right censoring is generated in discrete time. At the end of interval k, among subjects not yet randomly censored,

$$\Pr(C_{i,k+1} = 1 \mid C_{ik} = 0, H_{ik}) = \mathrm{expit}\{\gamma_0 + \gamma_L^\top L_i + \gamma_A A_{ik} + \gamma_X X_{ik} + \gamma_U U_{ik}\}.$$

If `param_cens = NULL`, no additional random censoring is generated. In all cases, the simulator also applies an administrative censoring time `C.a = study_end - T_e`, where `T_e` is sampled uniformly from the integers `0:floor(n_visit / 3)`. The observed time is the minimum of event time, random censoring time, and administrative censoring time.

The long-format output uses start-stop counting-process intervals. The terminal event time is kept as a continuous time when the event occurs inside an interval. If `lagk = TRUE`, the long-format data also include `Alag1-Alag3`, `Xlag1-Xlag3`, `Ulag1-Ulag3`, and one-step leads `Xnext1` and `Unext1`.

For the revised triplet-input `multimsm()` interface, the simulation output can be converted by defining `rand` as the subject-level baseline treatment `A.0`, `cross = S_CE`, and `subseq = pmax(S_ES, S_CS)`.

Value

A list with components:

- `dat`: Subject-level wide-format data. It contains `id`, selected baseline covariates, visit-indexed matrices expanded into columns for `X`, `U`, `A`, `V`, `S_ES`, `S_CE`, and `S_CS`, observed event/censoring information (`T.obs`, `D.obs`), observed switch time `T.w`, administrative censoring time `C.a`, and observed switch type `type_sw` (`1 = ES`, `2 = CE`, `3 = CS`, `NA = no observed switch before exit`).
- `dat_long`: Counting-process long-format data derived from `dat`, with one row per subject-interval while the subject is under observation. It includes `t.start`, `t.stop`, `event`, `C`, selected baseline covariates, `A`, `V`, `X`, `U`, and `S_ES`/`S_CE`/`S_CS`. If `lagk = TRUE`, it also includes the lag/lead variables described above.
- `C`: An `n` by `n_visit + 1` matrix containing the cumulative censoring process after combining random and administrative censoring.

- stats: Named summary statistics: init_trt, the proportion randomized to experimental treatment; switched, the proportion with an observed switch before exit; prop_ES, prop_CE, and prop_CS, the proportions of switch types among switchers; and prop_event, the observed event proportion.
- true_sdiff: If true_hr = TRUE and truth_n > 0, a list containing an approximate sustained-regime benchmark, including a fitted survival::survfit object, time grid t, survival curves surv0 and surv1, an approximate Cox hazard ratio cHR, survival difference surv_diff, and the simulated benchmark datasets dat_A0 and dat_A1. Otherwise NULL.

References

Keogh RH, Seaman SR, Gran JM, Vansteelandt S. Simulating longitudinal data from marginal structural models using the additive hazard model. *Biometrical Journal*. 2021;63:1526-1541.

Examples

```
set.seed(123)

sim_obj <- simswc(
  n = 200,
  n_visit = 8,
  base_cov = c("L1", "L3"),
  trt_prob = 0.5,
  param_tdcov  = c(-1.5, 0.3, 0.3, -0.2, 0.5),
  param_tdcov2 = c( 0.05, 0.2, 0.2, -0.2, 0.7),
  param_sw     = c(-3.0, 0.4, 0.4, -0.3, 0.3, 0.8),
  param_haz    = c(0.1, log(0.8), log(1.1), log(1.1),
                   log(1.4), log(1.4), log(1.3), log(1), log(0.9)),
  param_cens   = NULL,
  param_select = c(0, 0.5, 0.25),
  lagk = TRUE,
  true_hr = FALSE
)

head(sim_obj$dat_long)
sim_obj$stats

## Convert to the triplet interface used by multimsm().
dat <- sim_obj$dat_long
dat$rand <- stats::ave(dat$A, dat$id, FUN = function(x) rep(x[1], length(x)))
dat$cross <- dat$S_CE
dat$subseq <- pmax(dat$S_ES, dat$S_CS)

table(dat$rand, dat$cross, dat$subseq)
```

---

tcoarsen *Time coarsening and follow-up augmentation for start-stop survival data*

Description

`tcoarsen()` harmonizes irregular longitudinal survival data in start-stop form onto a user-specified discrete time grid. It maps observed update times to grid landmarks using either floor or ceiling coarsening, rebuilds start-stop intervals, and augments each individual's follow-up record by carrying the last known covariate and treatment values forward to empty grid intervals.

The function is intended as a transparent preprocessing utility for real-world clinical trial datasets with sparse or irregular follow-up times.The grid width and coarsening direction are substantive analysis choices and should usually be prespecified and examined in sensitivity analysis.

For use before `multimsm()`, include the triplet variables such as `rand`, `cross`, and `subseq` in `covs`, and set `absorb_vars = c("cross", "subseq")` when crossover and subsequent therapy indicators should be interpreted as absorbing current-status variables. The output remains an ordinary data frame that can be passed directly to downstream modeling functions.

Usage

```
tcoarsen(data, id, start, stop, event, covs = NULL, bin_width,
  dir_coarsen = c("floor", "ceiling"), origin = NULL, absorb_vars = NULL,
  keep_terminal_time = TRUE, gap_action = c("stop", "warn", "ignore"),
  add_visit = TRUE, visit_name = "visit", diagnostics = TRUE,
  verbose = TRUE)
```

Arguments

`data`	A `data.frame` in start-stop long format, with one or more rows per subject.
`id`	A character string for the subject identifier variable name.
`start`	A character string naming the interval start-time variable.
`stop`	A character string for the interval stop-time variable name.
`event`	A character string for the terminal event indicator. The event variable must be coded `0`/`1`, with at most one event per subject.
`covs`	Optional character vector of treatment or covariate variables to document as carried-forward predictors. Both baseline-only and time-varying variables are allowed. The function retains all original data columns, but `covs` is checked for existence and stored in the returned object's attributes for reproducibility. In practice, all **time-varying** covariates are encouraged to be included here.
`bin_width`	A positive numeric value giving the grid width in the same time units as `start` and `stop`, for example `30` for a 30-day grid.
`dir_coarsen`	Coarsening direction, either `"floor"` or `"ceiling"`. Floor coarsening moves updates backward to the start of the containing bin; ceiling coarsening moves updates forward to the end of the containing bin.
`origin`	Optional numeric grid origin. If `NULL`, the minimum observed start time is used. In trial applications this is often `0`.
`absorb_vars`	Optional character vector of variables to convert to absorbing status using within-subject cumulative maxima after sorting by time. These variables must be coded `0`/`1`. For multi-way switching data, this will often be `c("cross", "subseq")`.

keep_terminal_time
: Logical. If TRUE (default), each subject's exact event/censoring exit time is preserved as the final stop time. If FALSE, the exit time is also snapped to the selected grid. The default is strongly recommended for survival analyses.

gap_action
: How to handle gaps between adjacent within-subject start-stop intervals before coarsening. One of "stop", "warn", or "ignore". Overlapping intervals always stop with an error.

add_visit
: Logical. If TRUE (default), add a discrete grid index to the output.

visit_name
: A single character string giving the name of the visit-index variable to add when add_visit = TRUE. Defaults to "visit".

diagnostics
: Logical. If TRUE (default), attach preprocessing metadata and a by-visit row/event summary in diagnostics object of output.

verbose
: Logical. If TRUE, print a short preprocessing message.

Details

Suppose an individual has observed rows indexed by j, representing intervals [start_ij, stop_ij) over which the row values are assumed to apply. Given grid origin origin and bin width bin_width, an observed update time s is mapped to

- origin + floor((s - origin) / bin_width) * bin_width when dir_coarsen = "floor";
- origin + ceiling((s - origin) / bin_width) * bin_width when dir_coarsen = "ceiling".

After snapping update times, tcoarsen() inserts one row for each missing grid interval during which the subject remains under observation. Values of all carried variables are filled by last-known-value-carried-forward (LKCF). Terminal event/censoring times are preserved exactly by default, so an event at day 173 can remain stop = 173 even if the coarsening grid is monthly.

If multiple observed rows map to the same coarsened update time, the last observed row within that coarsened bin is kept. This rule is simple and reproducible, but it necessarily discards within-bin ordering information.

Value

A tcoarsen object with two components. The first component is a data.frame called dat_coarsen the time-coarsened input data with updated interval times and time-varying covariates under coarsened time grid; the second component is diagnostics for diagnostics settings and basic diagnostics.

References

Guerra SF, Schnitzer ME, Forget A, Blais L. Impact of discretization of the timeline for longitudinal causal inference methods. *Statistics in Medicine*. 2020 Sept;39(27):0277-6715.

See Also

multimsm()

Examples

```
if (requireNamespace("survival", quietly = TRUE)) {
  data("heart", package = "survival")

  heart_30 <- tcoarsen(
    data = heart,
    id = "id",
    start = "start",
    stop = "stop",
    event = "event",
    covs = c("transplant", "age", "surgery"),
    bin_width = 30,
    dir_coarsen = "floor",
    origin = 0,
    absorb_vars = "transplant",
    verbose = FALSE
  )

  utils::head(heart_30$dat_coarsen, 10)
}
```

Index