Package 'TTE'

August 28, 2026

Type Package

Title Design and Analysis Tools for Target Trial Emulation

Version 1.1.2

Date 2026-08-28

Maintainer Hisashi Noma <noma@ism.ac.jp>

Description Design and analysis tools for target trial emulation using longitudinal observational data. Functions are provided for checking person-period data, expanding longitudinal data into sequentially nested trials, estimating inverse probability weights for intention-to-treat and per-protocol analyses, and assessing weight distributions and covariate balance. Additional functions fit weighted pooled discrete-time outcome models, obtain standardized risks and treatment contrasts, and estimate weighted Kaplan-Meier and Aalen-Johansen curves. Two worked examples based on fully synthetic data illustrate an active-comparator new-user study comparing sodium-glucose cotransporter 2 inhibitors with dipeptidyl peptidase-4 inhibitors and an analysis of sequentially nested trials comparing angiotensin receptor blocker and calcium channel blocker strategies.

URL https://github.com/nomahi/TTE/

Depends R (>= 4.1.0)

Imports graphics, nnet, sandwich, stats, utils

Suggests testthat (>= 3.0.0)

License GPL-3

Encoding UTF-8

Config/testthat/edition 3

NeedsCompilation no

Author Hisashi Noma [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2520-9949>)

Repository CRAN

Date/Publication 2026-08-28 10:00:21 UTC

Contents

TTE-package . . . 2
ARB . . . 3
balance_wt . . . 5
boot_tte . . . 6
check_tte . . . 7
combine_wt . . . 8
curve_tte . . . 9
diagnose_wt . . . 10
discsurvreg . . . 11
est_wt . . . 13
seqdesign_tte . . . 14
SGLT2 . . . 16
std_tte . . . 17

Index **20**

`TTE-package` *Design and Analysis Tools for Target Trial Emulation*

Description

Design and analysis tools for target trial emulation using longitudinal observational data.

Details

The main workflow comprises the following steps:

- check person-period data with `check_tte()`;
- expand longitudinal data into sequentially nested trials with `seqdesign_tte()`;
- estimate and combine inverse probability weights with `est_wt()` and `combine_wt()`;
- assess covariate balance and diagnose weight distributions with `balance_wt()` and `diagnose_wt()`;
- fit weighted pooled discrete-time outcome models with `discsurvreg()`;
- obtain standardized risks and treatment contrasts with `std_tte()`; and
- estimate and plot weighted Kaplan-Meier and Aalen-Johansen curves with `curve_tte()`.

The package includes two pairs of fully synthetic datasets:

- `SGLT2` and `SGLT2_baseline`; and
- `ARB` and `ARB_baseline`.

No records from the motivating studies are included.

Author(s)

Hisashi Noma

References

Cashin, A. G., Hansford, H. J., Hernán, M. A., Swanson, S. A., Lee, H., Jones, M. D., Dahabreh, I. J., Dickerman, B. A., Egger, M., García-Albéniz, X., Golub, R. M., Islam, N., Lodi, S., Moreno-Betancur, M., Pearson, S. A., Schneeweiss, S., Sharp, M. K., Sterne, J. A. C., Stuart, E. A., and McAuley, J. H. (2025). Transparent reporting of observational studies emulating a target trial: the TARGET Statement. *BMJ* **390**, e087179.

Hernán, M. A., Dahabreh, I. J., Dickerman, B. A., and Swanson, S. A. (2025). The target trial framework for causal inference from observational data: why and when is it helpful? *Annals of Internal Medicine* **178**, 402–407.

Hernán, M. A., Wang, W., and Leaf, D. E. (2022). Target trial emulation: a framework for causal inference from observational data. *JAMA* **328**, 2446–2447.

Noma, H., Goto, A., Sugimoto, T., Sunada, H., Oda, F., Maeda, M., and Fukuda, H. (2026). Real-world effectiveness of SGLT2 inhibitors in adults aged 75 years or older: a target trial emulation. *Age and Ageing* **55**, afag246.

Noma, H., Kurita, N., Fukuma, S., Fujisawa, T., Oda, F., Maeda, M., and Fukuda, H. (2026). Heart failure and renal outcomes with angiotensin receptor blockers compared with calcium channel blockers in patients with chronic kidney disease: a target trial emulation. *Heart*. doi:10.1136/heartjnl2026328193.

Examples

```
library(TTE)
data(SGLT2)

fit <- discsurvreg(
  Y_death ~ A + splines::ns(time, df = 3) + trial_period,
  data = SGLT2,
  id = id,
  weights = w_itt,
  var_method = "standard"
)
fit
```

ARB *Synthetic ARB versus CCB Sequential-Trial Example*

Description

Fully synthetic baseline and person-month data representing sequentially nested new-user trials comparing angiotensin receptor blocker (ARB) and calcium channel blocker (CCB) strategies among people with chronic kidney disease. Heart failure hospitalization is the primary endpoint, and death is a competing event.

Usage

```
data(ARB_baseline)
data(ARB)
```

Format

`ARB_baseline` is a data frame with 900 person-trial entries and the following variables:

`id` Original individual identifier.

`trial` Identifier for the sequential trial in which the individual entered.

`A` Binary treatment-strategy indicator: 0 for CCB and 1 for ARB.

`treatment` Treatment-strategy label: `"CCB"` or `"ARB"`.

`age` Age at trial baseline, in years.

`female` Indicator for female sex: 0 for no and 1 for yes.

`bmi` Body mass index at trial baseline, in kg/m^2.

`sbp` Systolic blood pressure at trial baseline, in mm Hg.

`dbp` Diastolic blood pressure at trial baseline, in mm Hg.

`egfr` Estimated glomerular filtration rate at trial baseline, in mL/min/1.73 m^2.

`proteinuria` Indicator for proteinuria at trial baseline: 0 for no and 1 for yes.

`diabetes` Indicator for diabetes at trial baseline: 0 for no and 1 for yes.

`prior_heart_failure` Indicator for a history of heart failure at trial baseline: 0 for no and 1 for yes.

`prior_stroke` Indicator for a history of stroke at trial baseline: 0 for no and 1 for yes.

`trial_period` Calendar-period category for the trial baseline.

`ps_oracle` Probability of ARB initiation generated by the simulation treatment model.

`w_iptw` Stabilized baseline inverse probability of treatment weight.

`ARB` is a person-month data frame containing the variables in `ARB_baseline`, except `ps_oracle`, together with the following variables:

`time` Zero-based follow-up-month index; `time = 0` denotes the first month after trial baseline.

`Y_hf` Indicator of heart failure hospitalization during the interval: 0 for no and 1 for yes.

`Y_death` Indicator of death during the interval: 0 for no and 1 for yes.

`event_code` Competing-event code: 0 for no event, 1 for heart failure hospitalization, and 2 for death.

`stay_ltfu` Interval-specific indicator of remaining uncensored with respect to loss to follow-up: 0 for no and 1 for yes.

`stay_adherent` Interval-specific indicator of adherence to the baseline treatment strategy: 0 for no and 1 for yes.

`pp_at_risk` Indicator that the person-trial remains in the per-protocol risk set: 0 for no and 1 for yes.

`w_itt` Combined analysis weight for the intention-to-treat analysis.

`w_pp` Combined analysis weight for the per-protocol analysis.

In `ARB`, `sbp`, `dbp`, and `egfr` are updated over follow-up; the other clinical covariates are carried forward from trial baseline.

Details

The datasets were independently simulated and include no participant records from the motivating study. They are intended solely for education, software testing, and reproducible examples; numerical results obtained from them have no clinical interpretation.

References

Noma, H., Kurita, N., Fukuma, S., Fujisawa, T., Oda, F., Maeda, M., and Fukuda, H. (2026). Heart failure and renal outcomes with angiotensin receptor blockers compared with calcium channel blockers in patients with chronic kidney disease: a target trial emulation. *Heart*. doi:10.1136/heartjnl2026328193.

Examples

```
data(ARB_baseline)
data(ARB)
head(ARB_baseline)
table(ARB$event_code)
```

---

balance_wt *Assess Covariate Balance Before and After Weighting*

Description

Computes unweighted and weighted means, standard deviations, and standardized mean differences using externally supplied analysis weights.

Usage

```
balance_wt(formula, data, weights = NULL, absolute = TRUE)
```

Arguments

`formula`	A formula with a binary treatment variable on the left-hand side and baseline covariates on the right-hand side.
`data`	A data frame containing the variables in `formula`.
`weights`	Optional numeric analysis weights, the name of a weight column in `data`, or an object returned by `est_wt()` or `combine_wt()`.
`absolute`	Logical; whether to report standardized mean differences in absolute value.

Value

An object of class `balance_wt` containing unweighted and weighted covariate summaries and standardized mean differences. A `summary()` method provides a formatted balance table.

See Also

`est_wt`, `diagnose_wt`

Examples

```
data(SGLT2_baseline)
wt <- est_wt(
  A ~ age + female + bmi + hba1c + egfr + prior_heart_failure +
    prior_stroke + recent_hospitalization + trial_period,
  data = SGLT2_baseline, type = "treatment"
)
bal <- balance_wt(
  A ~ age + female + bmi + hba1c + egfr + prior_heart_failure +
    prior_stroke + recent_hospitalization + trial_period,
  data = SGLT2_baseline, weights = wt
)
summary(bal)
```

---

boot_tte *Individual-Cluster Bootstrap*

Description

Resamples original individuals while retaining all person-period rows and repeated trial entries belonging to each sampled individual.

Usage

```
boot_tte(data, id, statistic, R = 500L,
  conf_level = 0.95, seed = NULL, ...)
```

Arguments

`data`	Analysis data containing all rows required by `statistic`.
`id`	Original individual identifier used to define bootstrap clusters.
`statistic`	A function whose first argument is a bootstrap data frame and whose return value is a named numeric vector.
`R`	Number of bootstrap replicates.
`conf_level`	Confidence level for percentile intervals.
`seed`	Optional random-number seed.
`...`	Additional arguments passed to `statistic`.

Details

Sampling is performed at the level of the original individual. Consequently, all follow-up rows and all sequential-trial entries associated with a sampled individual are retained together.

Value

An object of class `boot_tte` containing the estimate from the original data, bootstrap replicates, and percentile confidence intervals.

Examples

```
data(SGLT2)
small <- subset(SGLT2, id <= 80 & time < 24)
stat_fun <- function(d) {
  fit <- discsurvreg(
    Y_death ~ A + time, data = d,
    id = id, weights = w_itt,
    var_method = "standard"
  )
  c(log_hr = unname(coef(fit)["A"]))
}
b <- boot_tte(small, id = id, statistic = stat_fun, R = 10, seed = 1)
b
```

`check_tte` *Check a Target-Trial Person-Period Dataset*

Description

Detects duplicate intervals, time-ordering problems, rows occurring after an event, within-trial treatment changes, and invalid analysis weights.

Usage

```
check_tte(data, id, time, trial = NULL, event = NULL,
  treatment = NULL, weights = NULL, expected_start = 0,
  allow_gaps = FALSE)
```

Arguments

`data`	A person-period data frame.
`id`	Original individual identifier.
`time`	Integer-valued time index since trial baseline.
`trial`	Optional trial identifier.
`event`	Optional binary event indicator or competing-event code.
`treatment`	Optional baseline treatment variable.
`weights`	Optional numeric analysis weights or the name of a weight column in `data`.
`expected_start`	Expected first value of the time index within each person-trial.
`allow_gaps`	Logical; whether gaps in integer time indices are allowed.

Value

An object of class `check_tte` containing the detected data-quality problems and the rows or person-trials affected by each problem.

See Also

seqdesign_tte

Examples

```
data(ARB)
chk <- check_tte(
  ARB,
  id = id, trial = trial, time = time,
  event = event_code, treatment = A, weights = w_itt
)
chk
```

---

combine_wt *Combine Inverse Probability Weight Components*

Description

Multiplies treatment, loss-to-follow-up, and adherence weight components and optionally truncates and normalizes the resulting analysis weights.

Usage

```
combine_wt(..., truncate = c(0.01, 0.99),
  normalize = c("none", "mean1", "sum_n"))
```

Arguments

...	Numeric vectors or objects returned by est_wt() or combine_wt(). All components must have equal length.
truncate	Lower and upper quantiles used to truncate the product. The default truncates at the 1st and 99th percentiles.
normalize	Normalization applied after truncation: "none" for no normalization, "mean1" for a mean of 1, or "sum_n" for a sum equal to the number of observations.

Value

An object of class combine_wt containing the combined weights and metadata describing truncation and normalization. Use weights() to extract the numeric weight vector.

See Also

est_wt, diagnose_wt

Examples

```
data(ARB)
w <- combine_wt(
  ARB$w_iptw,
  ARB$w_itt / ARB$w_iptw,
  normalize = "mean1"
)
summary(weights(w))
```

`curve_tte` *Estimate Weighted Kaplan-Meier or Aalen-Johansen Curves*

Description

Estimates weighted survival or cumulative-incidence curves from person-period risk sets. Curves are represented as right-continuous step functions.

Usage

```
curve_tte(data, time, event, treatment, weights = NULL,
  type = c("km", "aj"), cause = 1, id = NULL, trial = NULL,
  labels = NULL, bootstrap = 0L, conf_level = 0.95,
  seed = NULL)
```

Arguments

`data`	Person-period data with one at-risk row per interval.
`time`	Follow-up interval index.
`event`	Binary event indicator or competing-event code.
`treatment`	Treatment variable defining the curves to be compared.
`weights`	Optional numeric analysis weights, the name of a weight column in `data`, or an object returned by `est_wt()` or `combine_wt()`.
`type`	Curve type: `"km"` for Kaplan-Meier survival or `"aj"` for Aalen-Johansen cumulative incidence.
`cause`	Event code of interest for Aalen-Johansen estimation.
`id`	Original individual identifier, required when `bootstrap` > `0`.
`trial`	Optional trial identifier used together with `id` to define person-trial records.
`labels`	Optional display labels for the treatment groups.
`bootstrap`	Number of individual-cluster bootstrap replicates.
`conf_level`	Confidence level for percentile intervals.
`seed`	Optional random-number seed.

Details

The `time` variable indexes follow-up intervals, whereas summary and plotting times are expressed as elapsed follow-up time. For example, interval indices 0, ..., 59 represent the 60 intervals ending at elapsed times 1, ..., 60. Thus, `summary(x, time = 60)` reports the estimate after 60 follow-up intervals.

Value

An object of class `curve_tte` containing treatment-specific curve estimates and, when requested, bootstrap confidence intervals. A `summary()` method returns estimates at specified elapsed follow-up times.

See Also

`std_tte`, `boot_tte`

Examples

```
data(SGLT2)
km <- curve_tte(
  SGLT2, time = time, event = Y_death,
  treatment = treatment, weights = w_itt,
  type = "km", id = id, trial = trial
)
summary(km, time = 60)


data(ARB)
aj <- curve_tte(
  ARB, time = time, event = event_code,
  treatment = treatment, weights = w_itt,
  type = "aj", cause = 1, id = id, trial = trial
)
summary(aj, time = 60)
```

---

diagnose_wt *Diagnose Inverse Probability Weights*

Description

Summarizes weight distributions and effective sample sizes. When analysis data are supplied, treatment-group and risk-set diagnostics are also produced.

Usage

```
diagnose_wt(weights, data = NULL, treatment = NULL, time = NULL,
  id = NULL, trial = NULL, extreme = c(0.01, 0.99))
```

Arguments

weights	Numeric analysis weights, the name of a weight column in data, or an object returned by est_wt() or combine_wt().
data	Optional data frame containing variables used for stratified or risk-set diagnostics.
treatment	Optional treatment variable.
time	Optional follow-up interval index.
id	Optional original individual identifier.
trial	Optional trial identifier.
extreme	Lower and upper quantiles used to flag extreme weights.

Value

An object of class diagnose_wt containing distributional summaries, effective sample sizes, and any requested treatment-group or risk-set diagnostics.

See Also

est_wt, combine_wt, balance_wt

Examples

```
data(ARB)
diag <- diagnose_wt(
  w_itt, data = ARB,
  treatment = A, time = time, id = id, trial = trial
)
diag
```

---

discsurvreg *Fit a Weighted Pooled Discrete-Time Outcome Model*

Description

Fits a weighted pooled generalized linear model and computes cluster-robust inference with clustering at the level of the original individual.

Usage

```
discsurvreg(formula, data, id, weights = NULL,
  family = stats::quasibinomial(link = "cloglog"),
  var_method = c("standard", "MBN"), eform = TRUE,
  conf_level = 0.95)
```

Arguments

formula	Model formula for the interval-specific outcome probability.
data	Person-period data.
id	Original individual identifier used to define variance-estimation clusters.
weights	Optional numeric analysis weights, the name of a weight column in data, or an object returned by est_wt() or combine_wt().
family	A generalized linear model family. The default is a quasibinomial model with complementary log-log link.
var_method	Cluster-robust variance estimator: "standard" for the ordinary sandwich estimator or "MBN" for the Morel-Bokossa- Neerchal finite-sample correction.
eform	Logical; whether the default printed coefficient table reports exponentiated estimates.
conf_level	Confidence level used in printed and interval estimates.

Value

An object of class discsurvreg containing the fitted glm object, the cluster-robust covariance matrix, model coefficients, and analysis settings. Methods for coef(), vcov(), and confint() are available.

References

Morel, J. G., Bokossa, M. C., and Neerchal, N. K. (2003). Small sample correction for the variance of GEE estimators. *Biometrical Journal* **45**, 395–409. doi:10.1002/bimj.200390021.

See Also

std_tte

Examples

```
data(SGLT2)
fit <- discsurvreg(
  Y_death ~ A + splines::ns(time, df = 3) + trial_period,
  data = SGLT2,
  id = id, weights = w_itt,
  family = stats::quasibinomial(link = "cloglog"),
  var_method = "standard"
)
fit
confint(fit, parm = "A", eform = TRUE)
```

est_wt *Estimate Inverse Probability Weights*

Description

Estimates baseline treatment weights or cumulative censoring and adherence weights while retaining intermediate predicted probabilities and interval-specific weight factors.

Usage

```
est_wt(formula, data, numerator = NULL,
  type = c("treatment", "censoring", "adherence"),
  id = NULL, trial = NULL, time = NULL, cumulative = NULL,
  lag = NULL, stabilize = TRUE, truncate = c(0.01, 0.99),
  eps = 1e-06)
```

Arguments

formula	Denominator probability model.
data	A data frame containing the model variables.
numerator	Optional numerator probability model used to construct stabilized weights.
type	Weight type: "treatment", "censoring", or "adherence".
id	Original individual identifier, required for cumulative weights.
trial	Optional trial identifier used together with id to define person-trial records.
time	Follow-up interval index used to order rows before cumulative multiplication.
cumulative	Logical; whether interval-specific factors are multiplied within person-trial. When NULL, the setting is determined by type.
lag	Number of intervals by which cumulative weights are lagged before being assigned to outcome risk sets.
stabilize	Logical; whether numerator probabilities are used to construct stabilized weights.
truncate	Lower and upper quantiles used to truncate the final weights. The default truncates at the 1st and 99th percentiles.
eps	Lower probability bound used to avoid division by zero; predicted probabilities are bounded away from 0 and 1 by this amount.

Details

For type = "treatment", the function estimates a baseline inverse probability of treatment weight. For type = "censoring" or type = "adherence", interval-specific factors may be multiplied within person-trial to obtain cumulative weights. Weight truncation is an analysis choice; the default applies truncation at the 1st and 99th percentiles.

Value

An object of class `est_wt` containing the estimated weight vector, predicted numerator and denominator probabilities, interval-specific factors, and settings used in weight construction. The numeric weights can be extracted with `weights()`, and a `summary()` method provides diagnostics.

See Also

`combine_wt`, `diagnose_wt`, `balance_wt`

Examples

```
data(SGLT2_baseline)
wt <- est_wt(
  A ~ age + female + bmi + hba1c + egfr +
    prior_heart_failure + prior_stroke +
    recent_hospitalization + trial_period,
  data = SGLT2_baseline,
  type = "treatment"
)
summary(wt)


data(ARB)
wc <- est_wt(
  stay_ltfu ~ A + time + I(time^2) + age + female + egfr +
    prior_heart_failure,
  numerator = stay_ltfu ~ A + time + I(time^2),
  data = ARB,
  type = "censoring", id = id, trial = trial, time = time,
  cumulative = TRUE, lag = 1
)
summary(wc)
```

---

seqdesign_tte *Expand Longitudinal Data into Sequentially Nested Trials*

Description

Creates a trial entry at each eligible treatment decision time and expands each entry into follow-up person-period records.

Usage

```
seqdesign_tte(data, id, calendar_time, eligible, treatment,
  outcome, censor = NULL, max_follow = Inf, induction = 0L,
  keep = names(data))
```

Arguments

`data`	Longitudinal source data with one row per individual and calendar interval.
`id`	Original individual identifier.
`calendar_time`	Ordered calendar-time variable.
`eligible`	Logical eligibility indicator or expression evaluated at each potential trial baseline.
`treatment`	Treatment strategy observed at the eligible trial baseline.
`outcome`	Binary outcome indicator.
`censor`	Optional censoring indicator.
`max_follow`	Maximum number of retained follow-up intervals, including the interval indexed by time 0.
`induction`	Number of initial follow-up intervals excluded from outcome ascertainment.
`keep`	Names of source variables to retain in the expanded data.

Details

An individual may enter more than one sequential trial when eligible at multiple treatment decision times. Follow-up for each trial entry is obtained from subsequent rows of the original longitudinal data and ends according to the supplied outcome, censoring, and maximum-follow-up definitions.

Value

A person-period data frame with added `trial`, `time`, `A`, and `Y` variables, together with the retained source variables and indicators used to define follow-up.

See Also

`check_tte`

Examples

```
data(SGLT2)
source_data <- subset(SGLT2, id <= 8)
source_data$eligible <- source_data$time == 0
expanded <- seqdesign_tte(
  source_data, id = id, calendar_time = time,
  eligible = eligible, treatment = A, outcome = Y_death,
  max_follow = 24, induction = 1
)
head(expanded)
```

SGLT2 *Synthetic SGLT2i versus DPP-4i Target-Trial Example*

Description

Fully synthetic baseline and person-month data representing an active-comparator new-user target trial emulation comparing sodium-glucose cotransporter 2 inhibitors (SGLT2i) with dipeptidyl peptidase-4 inhibitors (DPP-4i) among older adults with type 2 diabetes. The primary endpoint is all-cause death after a one-month induction period.

Usage

```
data(SGLT2_baseline)
data(SGLT2)
```

Format

`SGLT2_baseline` is a data frame with 700 trial entries and the following variables:

`id` Original individual identifier.

`trial` Trial identifier.

`A` Binary treatment-strategy indicator: 0 for DPP-4i and 1 for SGLT2i.

`treatment` Treatment-strategy label: `"DPP-4i"` or `"SGLT2i"`.

`age` Age at trial baseline, in years.

`female` Indicator for female sex: 0 for no and 1 for yes.

`bmi` Body mass index at trial baseline, in kg/m^2.

`hba1c` Glycated hemoglobin at trial baseline, in percent.

`egfr` Estimated glomerular filtration rate at trial baseline, in mL/min/1.73 m^2.

`proteinuria` Indicator for proteinuria at trial baseline: 0 for no and 1 for yes.

`prior_heart_failure` Indicator for a history of heart failure at trial baseline: 0 for no and 1 for yes.

`prior_stroke` Indicator for a history of stroke at trial baseline: 0 for no and 1 for yes.

`recent_hospitalization` Indicator for a recent hospitalization at trial baseline: 0 for no and 1 for yes.

`trial_period` Calendar-period category for the trial baseline.

`ps_oracle` Probability of SGLT2i initiation generated by the simulation treatment model.

`w_iptw` Stabilized baseline inverse probability of treatment weight.

`SGLT2` is a person-month data frame containing the variables in `SGLT2_baseline`, except `ps_oracle`, together with the following variables:

`time` Zero-based follow-up-month index; `time = 0` denotes the first month after trial baseline.

`Y_death` Indicator of death during the interval: 0 for no and 1 for yes.

`event_code` Event code: 0 for no event and 1 for death.

`stay_ltfu` Interval-specific indicator of remaining uncensored with respect to loss to follow-up: 0 for no and 1 for yes.

`stay_adherent` Interval-specific indicator of adherence to the baseline treatment strategy: 0 for no and 1 for yes.

`pp_at_risk` Indicator that the trial entry remains in the per-protocol risk set: 0 for no and 1 for yes.

`w_itt` Combined analysis weight for the intention-to-treat analysis.

`w_pp` Combined analysis weight for the per-protocol analysis.

The baseline covariates are repeated across follow-up months in `SGLT2`.

Details

The datasets were independently simulated and include no participant records from the motivating study. They are intended solely for education, software testing, and reproducible examples; numerical results obtained from them have no clinical interpretation.

References

Noma, H., Goto, A., Sugimoto, T., Sunada, H., Oda, F., Maeda, M., and Fukuda, H. (2026). Real-world effectiveness of SGLT2 inhibitors in adults aged 75 years or older: a target trial emulation. *Age and Ageing* **55**, afag246.

Examples

```
data(SGLT2_baseline)
data(SGLT2)
head(SGLT2_baseline)
head(SGLT2)
```

`std_tte` *Standardize a Discrete-Time Model to Marginal Risks*

Description

Creates counterfactual copies of a target population, predicts interval-specific event probabilities, and obtains standardized survival, risk, cumulative incidence, risk differences, risk ratios, and time-specific numbers needed to treat or harm.

Usage

```
std_tte(fit, data, treatment, time, times,
  values = NULL, labels = NULL, competing_fit = NULL,
  target_weights = NULL)
```

Arguments

fit	A discsurvreg object for the event of interest.
data	Target population over which predictions are standardized.
treatment	Treatment variable used in the fitted model.
time	Follow-up interval index used in the fitted model.
times	Ordered interval-index values over which predictions are computed.
values	Treatment values defining the counterfactual strategies.
labels	Optional display labels for the treatment strategies.
competing_fit	Optional discsurvreg object for a competing event.
target_weights	Optional nonnegative weights used to average predictions over the target population.

Details

The values supplied in times index follow-up intervals, whereas summary horizons are expressed as elapsed follow-up time. For example, times = 0:59 represents the 60 intervals ending at elapsed times 1 through 60, so summary(x, horizon = 60) reports standardized estimates after 60 follow-up intervals.

When competing_fit is omitted, risks are obtained from the fitted event model. When a competing-event model is supplied, the function combines the event-specific probabilities to obtain standardized cumulative incidence.

Value

An object of class std_tte containing strategy-specific standardized curves and treatment contrasts. A summary() method returns estimates at a specified elapsed follow-up horizon.

See Also

discsurvreg, curve_tte

Examples

```
data(SGLT2)
data(SGLT2_baseline)
fit <- discsurvreg(
  Y_death ~ A + splines::ns(time, df = 3) + trial_period,
  data = SGLT2, id = id, weights = w_itt,
  var_method = "standard"
)
std <- std_tte(
  fit, data = SGLT2_baseline,
  treatment = A, time = time, times = 0:59,
  labels = c("DPP-4i", "SGLT2i")
)
summary(std, horizon = 60)
```

```
# Competing-risk example
data(ARB)
data(ARB_baseline)
fit_hf <- discsurvreg(
  Y_hf ~ A + splines::ns(time, df = 3) + trial_period,
  data = ARB, id = id, weights = w_itt,
  var_method = "standard"
)
fit_death <- discsurvreg(
  Y_death ~ A + splines::ns(time, df = 3) + trial_period,
  data = ARB, id = id, weights = w_itt,
  var_method = "standard"
)
std_hf <- std_tte(
  fit_hf, data = ARB_baseline,
  treatment = A, time = time, times = 0:59,
  competing_fit = fit_death, labels = c("CCB", "ARB")
)
summary(std_hf, horizon = 60)
```

Index

ARB, 3
ARB_baseline *(*ARB*)*, 3

balance_wt, 5, *11*, *14*
boot_tte, 6, *10*

check_tte, 7, *15*
combine_wt, 8, *11*, *14*
curve_tte, 9, *18*

diagnose_wt, *5*, *8*, 10, *14*
discsurvreg, 11, *18*

est_wt, *5*, *8*, *11*, 13

seqdesign_tte, *8*, 14
SGLT2, 16
SGLT2_baseline *(*SGLT2*)*, 16
std_tte, *10*, *12*, 17

TTE *(*TTE-package*)*, 2
TTE-package, 2