Package ‘TrialEmulation’

September 9, 2024

Title Causal Analysis of Observational Time-to-Event Data

Version 0.0.3.9

Description Implements target trial emulation methods to apply randomized clinical trial design and analysis in an observational setting. Using marginal structural models, it can estimate intention-to-treat and per-protocol effects in emulated trials using electronic health records. A description and application of the method can be found in Danaei et al (2013) <doi:10.1177/0962280211403603>.

License Apache License (>= 2)

URL `https://causal-lda.github.io/TrialEmulation/`, `https://github.com/Causal-LDA/TrialEmulation`

BugReports `https://github.com/Causal-LDA/TrialEmulation/issues`

Depends R (>= 3.5.0)

Imports Rcpp, broom (>= 0.7.10), checkmate, data.table (>= 1.9.8), formula.tools, mvtnorm, parglm, sandwich

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

LinkingTo Rcpp

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.2.9000

NeedsCompilation yes

Author Roonak Rezvani [aut] (<`https://orcid.org/0000-0001-5580-5058`>, Original package author),
Isaac Gravestock [aut, cre] (<`https://orcid.org/0000-0003-0283-2065`>),
Li Su [aut],
Medical Research Council (MRC) [fnd],
F. Hoffmann-La Roche AG [cph, fnd]

Maintainer Isaac Gravestock <isaac.gravestock@roche.com>

Repository CRAN

Date/Publication 2024-09-09 12:10:15 UTC

Contents

case_control_sampling_trials

Case-control sampling of expanded data for the sequence of emulated trials

Description

Perform case-control sampling of expanded data to create a data set of reduced size and calculate sampling weights to be used in `trial_msm()`.

Usage

```
case_control_sampling_trials(
  data_prep,
  p_control = NULL,
  subset_condition,
  sort = FALSE
)
```

Arguments

`data_prep` Result from `data_preparation()`.

`p_control` Control sampling probability for selecting potential controls at each follow-up time of each trial.

`subset_condition` Expression used to `subset()` the trial data before case-control sampling.

`sort` Sort data before applying case-control sampling to make sure that the resulting data are identical when sampling from the expanded data created with `separate_files = TRUE` or `separate_files = FALSE`.

Value

A `data.frame` or a `split()` `data.frame` if `length(p_control) > 1`. An additional column `sample_weight` containing the sample weights will be added to the result. These can be included in the models fit with `trial_msm()`.

Examples

```
# If necessary reduce the number of threads for data.table
data.table::setDTthreads(2)

data("te_data_ex")
samples <- case_control_sampling_trials(te_data_ex, p_control = 0.01)
```

`data_preparation` *Prepare data for the sequence of emulated target trials*

Description

This function expands observational data in the person-time format (i.e., the ‘long’ format) to emulate a sequence of target trials and also estimates the inverse probability of treatment and censoring weights as required.

Usage

```
data_preparation(
  data,
  id = "id",
  period = "period",
  treatment = "treatment",
  outcome = "outcome",
  eligible = "eligible",
  model_var = NULL,
  outcome_cov = ~1,
  estimand_type = c("ITT", "PP", "As-Treated"),
  switch_n_cov = ~1,
  switch_d_cov = ~1,
  first_period = NA,
  last_period = NA,
  use_censor_weights = FALSE,
  cense = NA,
  pool_cense = c("none", "both", "numerator"),
  cense_d_cov = ~1,
  cense_n_cov = ~1,
  eligible_wts_0 = NA,
  eligible_wts_1 = NA,
  where_var = NULL,
  data_dir,
  save_weight_models = FALSE,
  glm_function = "glm",
  chunk_size = 500,
  separate_files = FALSE,
  quiet = FALSE,
  ...
)
```

Arguments

`data` A `data.frame` containing all the required variables in the person-time format, i.e., the ‘long’ format.

`id` Name of the variable for identifiers of the individuals. Default is ‘id’.

`period` Name of the variable for the visit/period. Default is ‘period’.

`treatment` Name of the variable for the treatment indicator at that visit/period. Default is ‘treatment’.

`outcome` Name of the variable for the indicator of the outcome event at that visit/period. Default is ‘outcome’.

`eligible` Name of the variable for the indicator of eligibility for the target trial at that visit/period. Default is ‘eligible’.

`model_var` Treatment variables to be included in the marginal structural model for the emulated trials. `model_var = "assigned_treatment"` will create a variable `assigned_treatment` that is the assigned treatment at the trial baseline, typically used for ITT and per-protocol analyses. `model_var = "dose"` will create a variable `dose` that is the cumulative number of treatments received since the trial baseline, typically used in as-treated analyses.

`outcome_cov` A RHS formula with baseline covariates to be adjusted for in the marginal structural model for the emulated trials. Note that if a time-varying covariate is specified in `outcome_cov`, only its value at each of the trial baselines will be included in the expanded data.

`estimand_type` Specify the estimand for the causal analyses in the sequence of emulated trials. `estimand_type = "ITT"` will perform intention-to-treat analyses, where treatment switching after trial baselines are ignored. `estimand_type = "PP"` will perform per-protocol analyses, where individuals’ follow-ups are artificially censored and inverse probability of treatment weighting is applied. `estimand_type = "As-Treated"` will fit a standard marginal structural model for all possible treatment sequences, where individuals’ follow-ups are not artificially censored but treatment switching after trial baselines are accounted for by applying inverse probability of treatment weighting.

`switch_n_cov` A RHS formula to specify the logistic models for estimating the numerator terms of the inverse probability of treatment weights. A derived variable named `time_on_regime` containing the duration of time that the individual has been on the current treatment/non-treatment is available for use in these models.

`switch_d_cov` A RHS formula to specify the logistic models for estimating the denominator terms of the inverse probability of treatment weights.

`first_period` First time period to be set as trial baseline to start expanding the data.

`last_period` Last time period to be set as trial baseline to start expanding the data.

`use_censor_weights` Require the inverse probability of censoring weights. If `use_censor_weights = TRUE`, then the variable name of the censoring indicator needs to be provided in the argument `cense`.

`cense` Variable name for the censoring indicator. Required if `use_censor_weights = TRUE`.

`pool_cense` Fit pooled or separate censoring models for those treated and those untreated at the immediately previous visit. Pooling can be specified for the models for the numerator and denominator terms of the inverse probability of censoring weights. One of `"none"`, `"numerator"`, or `"both"` (default is `"none"` except when `estimand_type = "ITT"` then default is `"numerator"`).

`cense_d_cov` A RHS formula to specify the logistic models for estimating the denominator terms of the inverse probability of censoring weights.

`cense_n_cov` A RHS formula to specify the logistic models for estimating the numerator terms of the inverse probability of censoring weights.

`eligible_wts_0` See definition for `eligible_wts_1`

`eligible_wts_1` Exclude some observations when fitting the models for the inverse probability of treatment weights. For example, if it is assumed that an individual will stay on treatment for at least 2 visits, the first 2 visits after treatment initiation by definition have a probability of staying on the treatment of 1.0 and should thus be excluded from the weight models for those who are on treatment at the immediately previous visit. Users can define a variable that indicates that these 2 observations are ineligible for the weight model for those who are on treatment at the immediately previous visit and add the variable name in the argument `eligible_wts_1`. Similar definitions are applied to `eligible_wts_0` for excluding observations when fitting the models for the inverse probability of treatment weights for those who are not on treatment at the immediately previous visit.

`where_var` Specify the variable names that will be used to define subgroup conditions when fitting the marginal structural model for a subgroup of individuals. Need to specify jointly with the argument `where_case`.

`data_dir` Directory to save model objects when `save_weight_models=TRUE` and expanded data as separate CSV files names as `trial_i.csv`s if `separate_files = TRUE`. If the specified directory does not exist it will be created. If the directory already contains trial files, an error will occur, other files may be overwritten.

`save_weight_models` Save model objects for estimating the weights in `data_dir`.

`glm_function` Specify which glm function to use for the marginal structural model from the `stats` or `parglm` packages. The default function is the `glm` function in the `stats` package. Users can also specify `glm_function = "parglm"` such that the `parglm` function in the `parglm` package can be used for fitting generalized linear models in parallel. The default control setting for `parglm` is `nthreads = 4` and `method = "FAST"`, where four cores and Fisher information are used for faster computation. Users can change the default control setting by passing the arguments `nthreads` and `method` in the `parglm.control` function of the `parglm` package, or alternatively, by passing a `control` argument with a list produced by `parglm.control(nthreads = , method = )`.

`chunk_size` Number of individuals whose data to be processed in one chunk when `separate_files = TRUE`

`separate_files` Save expanded data in separate CSV files for each trial.

`quiet` Suppress the printing of progress messages and summaries of the fitted models.

`...` Additional arguments passed to `glm_function`. This may be used to specify initial values of parameters or arguments to `control`. See stats::glm, parglm::parglm and `parglm::parglm.control()` for more information.

Details

The arguments `chunk_size` and `separate_files` allow for processing of large datasets that would not fit in memory once expanded. When `separate_files = TRUE`, the input data are processed in chunks of individuals and saved into separate files for each emulated trial. These separate files can be sampled by case-control sampling to create a reduced dataset for the modelling.

Value

An object of class `TE_data_prep`, which can either be sampled from (case_control_sampling_trials) or directly used in a model (trial_msm). It contains the elements

data the expanded dataset for all emulated trials. If `separate_files = FALSE`, it is a `data.table`; if `separate_files = TRUE`, it is a character vector with the file path of the expanded data as CSV files.

min_period index for the first trial in the expanded data

max_period index for the last trial in the expanded data

N the total number of observations in the expanded data

data_template a zero-row `data.frame` with the columns and attributes of the expanded data

switch_models a list of summaries of the models fitted for inverse probability of treatment weights, if `estimand_type` is `"PP"` or `"As-Treated"`

censor_models a list of summaries of the models fitted for inverse probability of censoring weights, if `use_censor_weights=TRUE`

args a list contain the parameters used to prepare the data and fit the weight models

initiators *A wrapper function to perform data preparation and model fitting in a sequence of emulated target trials*

Description

An all-in-one analysis using a sequence of emulated target trials. This provides a simplified interface to the main functions `data_preparation()` and `trial_msm()`.

Usage

```
initiators(
  data,
  id = "id",
  period = "period",
  treatment = "treatment",
  outcome = "outcome",
  eligible = "eligible",
  outcome_cov = ~1,
  estimand_type = c("ITT", "PP", "As-Treated"),
  model_var = NULL,
  switch_n_cov = ~1,
  switch_d_cov = ~1,
  first_period = NA,
  last_period = NA,
  first_followup = NA,
  last_followup = NA,
  use_censor_weights = FALSE,
  save_weight_models = FALSE,
  analysis_weights = c("asis", "unweighted", "p99", "weight_limits"),
  weight_limits = c(0, Inf),
  cense = NA,
  pool_cense = c("none", "both", "numerator"),
  cense_d_cov = ~1,
  cense_n_cov = ~1,
  include_followup_time = ~followup_time + I(followup_time^2),
  include_trial_period = ~trial_period + I(trial_period^2),
  eligible_wts_0 = NA,
  eligible_wts_1 = NA,
  where_var = NULL,
  where_case = NA,
  data_dir,
  glm_function = "glm",
  quiet = FALSE,
  ...
)
```

Arguments

`data` A `data.frame` containing all the required variables in the person-time format, i.e., the ‘long’ format.

`id` Name of the variable for identifiers of the individuals. Default is ‘id’.

`period` Name of the variable for the visit/period. Default is ‘period’.

`treatment` Name of the variable for the treatment indicator at that visit/period. Default is ‘treatment’.

`outcome` Name of the variable for the indicator of the outcome event at that visit/period. Default is ‘outcome’.

`eligible` Name of the variable for the indicator of eligibility for the target trial at that visit/period. Default is ‘eligible’.

`outcome_cov` A RHS formula with baseline covariates to be adjusted for in the marginal structural model for the emulated trials. Note that if a time-varying covariate is specified in `outcome_cov`, only its value at each of the trial baselines will be included in the expanded data.

`estimand_type` Specify the estimand for the causal analyses in the sequence of emulated trials. `estimand_type = "ITT"` will perform intention-to-treat analyses, where treatment switching after trial baselines are ignored. `estimand_type = "PP"` will perform per-protocol analyses, where individuals’ follow-ups are artificially censored and inverse probability of treatment weighting is applied. `estimand_type = "As-Treated"` will fit a standard marginal structural model for all possible treatment sequences, where individuals’ follow-ups are not artificially censored but treatment switching after trial baselines are accounted for by applying inverse probability of treatment weighting.

`model_var` Treatment variables to be included in the marginal structural model for the emulated trials. `model_var = "assigned_treatment"` will create a variable `assigned_treatment` that is the assigned treatment at the trial baseline, typically used for ITT and per-protocol analyses. `model_var = "dose"` will create a variable `dose` that is the cumulative number of treatments received since the trial baseline, typically used in as-treated analyses.

`switch_n_cov` A RHS formula to specify the logistic models for estimating the numerator terms of the inverse probability of treatment weights. A derived variable named `time_on_regime` containing the duration of time that the individual has been on the current treatment/non-treatment is available for use in these models.

`switch_d_cov` A RHS formula to specify the logistic models for estimating the denominator terms of the inverse probability of treatment weights.

`first_period` First time period to be set as trial baseline to start expanding the data.

`last_period` Last time period to be set as trial baseline to start expanding the data.

`first_followup` First follow-up time/visit in the trials to be included in the marginal structural model for the outcome event.

`last_followup` Last follow-up time/visit in the trials to be included in the marginal structural model for the outcome event.

`use_censor_weights`
: Require the inverse probability of censoring weights. If `use_censor_weights = TRUE`, then the variable name of the censoring indicator needs to be provided in the argument `cense`.

`save_weight_models`
: Save model objects for estimating the weights in `data_dir`.

`analysis_weights`
: Choose which type of weights to be used for fitting the marginal structural model for the outcome event.

  - `"asis"`: use the weights as calculated.
  - `"p99"`: use weights truncated at the 1st and 99th percentiles (based on the distribution of weights in the entire sample).
  - `"weight_limits"`: use weights truncated at the values specified in `weight_limits`.
  - `"unweighted"`: set all analysis weights to 1, even if treatment weights or censoring weights were calculated.

`weight_limits`
: Lower and upper limits to truncate weights, given as `c(lower, upper)`

`cense`
: Variable name for the censoring indicator. Required if `use_censor_weights = TRUE`.

`pool_cense`
: Fit pooled or separate censoring models for those treated and those untreated at the immediately previous visit. Pooling can be specified for the models for the numerator and denominator terms of the inverse probability of censoring weights. One of `"none"`, `"numerator"`, or `"both"` (default is `"none"` except when `estimand_type = "ITT"` then default is `"numerator"`).

`cense_d_cov`
: A RHS formula to specify the logistic models for estimating the denominator terms of the inverse probability of censoring weights.

`cense_n_cov`
: A RHS formula to specify the logistic models for estimating the numerator terms of the inverse probability of censoring weights.

`include_followup_time`
: The model to include the follow up time/visit of the trial (`followup_time`) in the marginal structural model, specified as a RHS formula.

`include_trial_period`
: The model to include the trial period (`trial_period`) in the marginal structural model, specified as a RHS formula.

`eligible_wts_0`
: See definition for `eligible_wts_1`

`eligible_wts_1`
: Exclude some observations when fitting the models for the inverse probability of treatment weights. For example, if it is assumed that an individual will stay on treatment for at least 2 visits, the first 2 visits after treatment initiation by definition have a probability of staying on the treatment of 1.0 and should thus be excluded from the weight models for those who are on treatment at the immediately previous visit. Users can define a variable that indicates that these 2 observations are ineligible for the weight model for those who are on treatment at the immediately previous visit and add the variable name in the argument `eligible_wts_1`. Similar definitions are applied to `eligible_wts_0` for excluding observations when fitting the models for the inverse probability of treatment weights for those who are not on treatment at the immediately previous visit.

`where_var` Specify the variable names that will be used to define subgroup conditions when fitting the marginal structural model for a subgroup of individuals. Need to specify jointly with the argument `where_case`.

`where_case` Define conditions using variables specified in `where_var` when fitting a marginal structural model for a subgroup of the individuals. For example, if `where_var= "age"`, `where_case = "age >= 30"` will only fit the marginal structural model to the subgroup of individuals. who are 30 years old or above.

`data_dir` Directory to save model objects in.

`glm_function` Specify which glm function to use for the marginal structural model from the `stats` or `parglm` packages. The default function is the `glm` function in the `stats` package. Users can also specify `glm_function = "parglm"` such that the `parglm` function in the `parglm` package can be used for fitting generalized linear models in parallel. The default control setting for `parglm` is `nthreads = 4` and `method = "FAST"`, where four cores and Fisher information are used for faster computation. Users can change the default control setting by passing the arguments `nthreads` and `method` in the `parglm.control` function of the `parglm` package, or alternatively, by passing a `control` argument with a list produced by `parglm.control(nthreads = , method = )`.

`quiet` Suppress the printing of progress messages and summaries of the fitted models.

`...` Additional arguments passed to `glm_function`. This may be used to specify initial values of parameters or arguments to `control`. See stats::glm, parglm::parglm and `parglm::parglm.control()` for more information.

Value

Returns the result of `trial_msm()` from the expanded data. An object of class `TE_msm` containing

model a `glm` object

robust a list containing a summary table of estimated regression coefficients and the robust covariance matrix

---

`predict.TE_msm` *Predict marginal cumulative incidences with confidence intervals for a target trial population*

Description

This function predicts the marginal cumulative incidences when a target trial population receives either the treatment or non-treatment at baseline (for an intention-to-treat analysis) or either sustained treatment or sustained non-treatment (for a per-protocol analysis). The difference between these cumulative incidences is the estimated causal effect of treatment. Currently, the `predict` function only provides marginal intention-to-treat and per-protocol effects, therefore it is only valid when `estimand_type = "ITT"` or `estimand_type = "PP"`.

Usage

```
## S3 method for class 'TE_msm'
predict(
  object,
  newdata,
  predict_times,
  conf_int = TRUE,
  samples = 100,
  type = c("cum_inc", "survival"),
  ...
)
```

Arguments

`object` Object from `trial_msm()` or `initiators()`.

`newdata` Baseline trial data that characterise the target trial population that marginal cumulative incidences or survival probabilities are predicted for. `newdata` must have the same columns and formats of variables as in the fitted marginal structural model specified in `trial_msm()` or `initiators()`. If `newdata` contains rows with `followup_time` > `0` these will be removed.

`predict_times` Specify the follow-up visits/times where the marginal cumulative incidences or survival probabilities are predicted.

`conf_int` Construct the point-wise 95-percent confidence intervals of cumulative incidences for the target trial population under treatment and non-treatment and their differences by simulating the parameters in the marginal structural model from a multivariate normal distribution with the mean equal to the marginal structural model parameter estimates and the variance equal to the estimated robust covariance matrix.

`samples` Number of samples used to construct the simulation-based confidence intervals.

`type` Specify cumulative incidences or survival probabilities to be predicted. Either cumulative incidence (`"cum_inc"`) or survival probability (`"survival"`).

`...` Further arguments passed to or from other methods.

Value

A list of three data frames containing the cumulative incidences for each of the assigned treatment options (treatment and non-treatment) and the difference between them.

Examples

```
# If necessary set the number of `data.table` threads
data.table::setDTthreads(2)

data("te_model_ex")
predicted_ci <- predict(te_model_ex, predict_times = 0:30, samples = 10)

# Plot the cumulative incidence curves under treatment and non-treatment
```

```
plot(predicted_ci[[1]]$followup_time, predicted_ci[[1]]$cum_inc,
  type = "l",
  xlab = "Follow-up Time", ylab = "Cumulative Incidence",
  ylim = c(0, 0.7)
)
lines(predicted_ci[[1]]$followup_time, predicted_ci[[1]]$`2.5%`, lty = 2)
lines(predicted_ci[[1]]$followup_time, predicted_ci[[1]]$`97.5%`, lty = 2)

lines(predicted_ci[[2]]$followup_time, predicted_ci[[2]]$cum_inc, type = "l", col = 2)
lines(predicted_ci[[2]]$followup_time, predicted_ci[[2]]$`2.5%`, lty = 2, col = 2)
lines(predicted_ci[[2]]$followup_time, predicted_ci[[2]]$`97.5%`, lty = 2, col = 2)
legend("topleft", title = "Assigned Treatment", legend = c("0", "1"), col = 1:2, lty = 1)

# Plot the difference in cumulative incidence over follow up
plot(predicted_ci[[3]]$followup_time, predicted_ci[[3]]$cum_inc_diff,
  type = "l",
  xlab = "Follow-up Time", ylab = "Difference in Cumulative Incidence",
  ylim = c(0.0, 0.5)
)
lines(predicted_ci[[3]]$followup_time, predicted_ci[[3]]$`2.5%`, lty = 2)
lines(predicted_ci[[3]]$followup_time, predicted_ci[[3]]$`97.5%`, lty = 2)
```

print.TE_weight_summary

Print a weight summary object

Description

Print a weight summary object

Usage

```
## S3 method for class 'TE_weight_summary'
print(x, full = TRUE, ...)
```

Arguments

x	print TE_weight_summary object.
full	Print full or short summary.
...	Arguments passed to print.data.frame.

Value

No return value, only for printing.

`summary.TE_data_prep` *Summary methods*

Description

Print summaries of data and model objects produced by `TrialEmulation`.

Usage

```
## S3 method for class 'TE_data_prep'
summary(object, ...)

## S3 method for class 'TE_data_prep_sep'
summary(object, ...)

## S3 method for class 'TE_data_prep_dt'
summary(object, ...)

## S3 method for class 'TE_msm'
summary(object, ...)

## S3 method for class 'TE_robust'
summary(object, ...)
```

Arguments

`object`	Object to print summary
`...`	Additional arguments passed to print methods.

Value

No value, displays summaries of object.

`te_data_ex` *Example of a prepared data object*

Description

A small example object from data_preparation used in examples. It is created with the following code:

Usage

```
te_data_ex
```

Format

An object of class TE_data_prep_dt (inherits from TE_data_prep) of length 6.

Details

```
dat <- trial_example[trial_example$id < 200, ]

te_data_ex <- data_preparation(
data = dat,
 outcome_cov = c("nvarA", "catvarA"),
 first_period = 260,
 last_period = 280
)
```

See Also

te_model_ex

---

te_model_ex *Example of a fitted marginal structural model object*

Description

A small example object from trial_msm used in examples. It is created with the following code:

Usage

```
te_model_ex
```

Format

An object of class TE_msm of length 3.

Details

```
te_model_ex <- trial_msm(
 data = data_subset,
 outcome_cov = c("catvarA", "nvarA"),
 last_followup = 40,
 model_var = "assigned_treatment",
 include_followup_time = ~followup_time,
 include_trial_period = ~trial_period,
 use_sample_weights = FALSE,
 quiet = TRUE,
 glm_function = "glm"
 )
```

See Also

te_data_ex

trial_example *Example of longitudinal data for sequential trial emulation*

Description

A dataset containing the treatment, outcomes and other attributes of 503 patients for sequential trial emulation. See `vignette("Getting-Started")`.

Usage

```
trial_example
```

Format

A data frame with 48400 rows and 11 variables:

id patient identifier

eligible eligible for trial start in this period, 1=yes, 0=no

period time period

outcome indicator for outcome in this period, 1=event occurred, 0=no event

treatment indicator for receiving treatment in this period, 1=treatment, 0=no treatment

catvarA A categorical variable relating to treatment and the outcome

catvarB A categorical variable relating to treatment and the outcome

catvarC A categorical variable relating to treatment and the outcome

nvarA A numerical variable relating to treatment and the outcome

nvarB A numerical variable relating to treatment and the outcome

nvarC A numerical variable relating to treatment and the outcome

trial_msm *Fit the marginal structural model for the sequence of emulated trials*

Description

Apply a weighted pooled logistic regression to fit the marginal structural model for the sequence of emulated trials and calculates the robust covariance matrix of parameter using the sandwich estimator.

Usage

```
trial_msm(
  data,
  outcome_cov = ~1,
  estimand_type = c("ITT", "PP", "As-Treated"),
  model_var = NULL,
  first_followup = NA,
  last_followup = NA,
  analysis_weights = c("asis", "unweighted", "p99", "weight_limits"),
  weight_limits = c(0, Inf),
  include_followup_time = ~followup_time + I(followup_time^2),
  include_trial_period = ~trial_period + I(trial_period^2),
  where_case = NA,
  glm_function = c("glm", "parglm"),
  use_sample_weights = TRUE,
  quiet = FALSE,
  ...
)
```

Arguments

`data`	A `data.frame` containing all the required variables in the person-time format, i.e., the 'long' format.
`outcome_cov`	A RHS formula with baseline covariates to be adjusted for in the marginal structural model for the emulated trials. Note that if a time-varying covariate is specified in `outcome_cov`, only its value at each of the trial baselines will be included in the expanded data.
`estimand_type`	Specify the estimand for the causal analyses in the sequence of emulated trials. `estimand_type = "ITT"` will perform intention-to-treat analyses, where treatment switching after trial baselines are ignored. `estimand_type = "PP"` will perform per-protocol analyses, where individuals' follow-ups are artificially censored and inverse probability of treatment weighting is applied. `estimand_type = "As-Treated"` will fit a standard marginal structural model for all possible treatment sequences, where individuals' follow-ups are not artificially censored but treatment switching after trial baselines are accounted for by applying inverse probability of treatment weighting.

`model_var` Treatment variables to be included in the marginal structural model for the emulated trials. `model_var = "assigned_treatment"` will create a variable `assigned_treatment` that is the assigned treatment at the trial baseline, typically used for ITT and per-protocol analyses. `model_var = "dose"` will create a variable `dose` that is the cumulative number of treatments received since the trial baseline, typically used in as-treated analyses.

`first_followup` First follow-up time/visit in the trials to be included in the marginal structural model for the outcome event.

`last_followup` Last follow-up time/visit in the trials to be included in the marginal structural model for the outcome event.

`analysis_weights`

Choose which type of weights to be used for fitting the marginal structural model for the outcome event.

- `"asis"`: use the weights as calculated.
- `"p99"`: use weights truncated at the 1st and 99th percentiles (based on the distribution of weights in the entire sample).
- `"weight_limits"`: use weights truncated at the values specified in `weight_limits`.
- `"unweighted"`: set all analysis weights to 1, even if treatment weights or censoring weights were calculated.

`weight_limits` Lower and upper limits to truncate weights, given as `c(lower, upper)`

`include_followup_time`

The model to include the follow up time/visit of the trial (`followup_time`) in the marginal structural model, specified as a RHS formula.

`include_trial_period`

The model to include the trial period (`trial_period`) in the marginal structural model, specified as a RHS formula.

`where_case` Define conditions using variables specified in `where_var` when fitting a marginal structural model for a subgroup of the individuals. For example, if `where_var= "age"`, `where_case = "age >= 30"` will only fit the marginal structural model to the subgroup of individuals. who are 30 years old or above.

`glm_function` Specify which glm function to use for the marginal structural model from the `stats` or `parglm` packages. The default function is the `glm` function in the `stats` package. Users can also specify `glm_function = "parglm"` such that the `parglm` function in the `parglm` package can be used for fitting generalized linear models in parallel. The default control setting for `parglm` is `nthreads = 4` and `method = "FAST"`, where four cores and Fisher information are used for faster computation. Users can change the default control setting by passing the arguments `nthreads` and `method` in the `parglm.control` function of the `parglm` package, or alternatively, by passing a `control` argument with a list produced by `parglm.control(nthreads = , method = )`.

`use_sample_weights`

Use case-control sampling weights in addition to inverse probability weights for treatment and censoring. `data` must contain a column `sample_weight`. The final weights used in the pooled logistic regression are calculated as `weight = weight * sample_weight`.

`quiet`	Suppress the printing of progress messages and summaries of the fitted models.
`...`	Additional arguments passed to `glm_function`. This may be used to specify initial values of parameters or arguments to `control`. See stats::glm, parglm::parglm and `parglm::parglm.control()` for more information.

Details

The model formula is constructed by combining the arguments `outcome_cov`, `model_var`, `include_followup_time`, and `include_trial_period`.

Value

Object of class `TE_msm` containing

model a `glm` object

robust a list containing a summary table of estimated regression coefficients and the robust covariance matrix

args a list contain the parameters used to prepare and fit the model

`vignette_switch_data` *Example of expanded longitudinal data for sequential trial emulation*

Description

This is the expanded dataset created in the `vignette("Getting-Started")` known as `switch_data`.

Usage

```
vignette_switch_data
```

Format

A data frame with 1939053 rows and 7 variables:

id patient identifier

trial_period trial start time period

followup_time follow up time within trial

outcome indicator for outcome in this period, 1=event occurred, 0=no event

treatment indicator for receiving treatment in this period, 1=treatment, 0=non-treatment

assigned_treatment indicator for assigned treatment at baseline of the trial, 1=treatment, 0=non-treatment

weight weights for use with model fitting

catvarA A categorical variable relating to treatment and the outcome

catvarB A categorical variable relating to treatment and the outcome

catvarC A categorical variable relating to treatment and the outcome

nvarA A numerical variable relating to treatment and the outcome

nvarB A numerical variable relating to treatment and the outcome

nvarC A numerical variable relating to treatment and the outcome

Index