Package ‘trtswitch’

October 6, 2025

Type Package

Title Treatment Switching

Version 0.2.0

Description Implements rank-preserving structural failure time model (RPSFTM), iterative parameter estimation (IPE), inverse probability of censoring weights (IPCW), marginal structural model (MSM), simple two-stage estimation (TSEsimp), and improved two-stage with g-estimation (TSEgest) methods for treatment switching in randomized clinical trials.

License GPL (>= 2)

Imports Rcpp (>= 1.0.9), data.table (>= 1.14.10)

LinkingTo Rcpp

Suggests testthat (>= 3.0.0), dplyr (>= 1.1.4), tidyr (>= 1.3.1),
survival (>= 3.5-7), ggplot2 (>= 3.3.6), knitr (>= 1.45),
rmarkdown (>= 2.25)

VignetteBuilder knitr

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation yes

Depends R (>= 2.10)

LazyData true

URL `https://github.com/kaifenglu/trtswitch`

BugReports `https://github.com/kaifenglu/trtswitch/issues`

Config/testthat/edition 3

Author Kaifeng Lu [aut, cre] (ORCID: <`https://orcid.org/0000-0002-6160-7119`>)

Maintainer Kaifeng Lu <`kaifenglu@gmail.com`>

Repository CRAN

Date/Publication 2025-10-06 07:20:02 UTC

Contents

trtswitch-package *Treatment Switching*

Description

Implements rank-preserving structural failure time model (RPSFTM), iterative parameter estimation (IPE), inverse probability of censoring weights (IPCW), marginal structural model (MSM), simple two-stage estimation (TSEsimp), and improved two-stage with g-estimation (TSEgest) methods for treatment switching in randomized clinical trials.

Details

To enable bootstrapping of the parameter estimates, we implements many standard survival analysis methods in C++. These include but are not limited to Kaplan-Meier estimates of the survival curves, log-rank tests, accelerate failure time models, and Cox proportional hazards models.

All treatment switching adjustment methods allow treatment switching in both treatment arms, stratification and covariates adjustment. For the AFT models, stratification factors are included as covariates (main effects only or all-way interactions) because SAS PROC LIFEREG does not have the strata statement. The RPSFTM, IPE and TSE methods can be used with or without recensoring. The IPCW and MSM methods can produce stabilized and truncated weights.

The `treat` variable adopts a treatment-before-control order, except with 1/0 or TRUE/FALSE coding.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

James M. Robins and Anastasios A. Tsiatis. Correcting for non-compliance in randomized trials using rank preserving structural failure time models. Communications in Statistics. 1991;20(8):2609-2631.

Ian R. White, Adbel G. Babiker, Sarah Walker, and Janet H. Darbyshire. Randomization-based methods for correcting for treatment changes: Examples from the CONCORDE trial. Statistics in Medicine. 1999;18:2617-2634.

Michael Branson and John Whitehead. Estimating a treatment effect in survival studies in which patients switch treatment. Statistics in Medicine. 2002;21:2449-2463.

Ian R White. Letter to the Editor: Estimating treatment effects in randomized trials with treatment switching. Statistics in Medicine. 2006;25:1619-1622.

James M. Robins and Dianne M. Finkelstein. Correcting for noncompliance and dependent censoring in an AIDS clinical trial with inverse probability of censoring weighted (IPCW) log-rank tests. Biometrics. 2000;56:779-788.

Nicholas R. Latimer, Keith R. Abrams, Paul C. Lambert, Michael K. Crowther, Allan J. Wailoo, Jonathan P. Morden, Ron L. Akehurst, and Michael J. Campbell. Adjusting for treatment switching in randomised controlled trials - A simulation study and a simplified two-stage method. Statistical Methods in Medical Research. 2017;26(2):724-751.

Nicholas R. Latimer, Ian R. White, Kate Tilling, and Ulrike Siebert. Improved two-stage estimation to adjust for treatment switching in randomised trials: g-estimation to address time-dependent confounding. Statistical Methods in Medical Research. 2020;29(10):2900-2918.

James M. Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. Epidemiology. 2000;11(5):550-560.

Miguel Angel Hernan, Babette Brumback, and James M. Robins. Marginal structural modesl to estimate the causual effect of zidovudine on the survival of HIV-positive men. Epidemiology. 2000;11(5):561-570.

`adsl` *Baseline subject-level data*

Description

This data set contains baseline subject-level data. Of note, `PDDT` can be derived from the `ADT` variable of the `ADTTE` data set by selecting `PARAMCD == "INPFS" & CNSR == 0 & EVNTDESC == "PROGRESSIVE DISEASE"`. Additionally, `OSDT` and `DIED` can be derived from the `ADT` and `CNSR` variables of the `ADTTE` data set by selecting `PARAMCD == "OS"`.

Usage

```
adsl
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 412 rows and 12 columns.

Details

`SUBJID` subject ID

`SEX` sex: "M" or "F"

`STRAT1V` stratification factor 1: ECOG PS

`STRAT2V` stratification factor 2: inv. chosen chemotherapy

`RANDDT` randomization date

`TRT01P` planned treatment: Active or Placebo

`TRTSDT` treatment start date

`PDDT` date of disease progression

`XODT` date of treatment crossover

`OSDT` date of death or censoring

`DIED` whether the patient died

`DCUTDT` date of data cut

adtdc *Longitudinal time-dependent covariate data*

Description

This data set contains longitudinal time-dependent covariate data on ECOG101 and LDH.

Usage

```
adtdc
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9813 rows and 4 columns.

Details

`SUBJID` subject ID

`PARAMCD` parameter code

`ADT` analysis date

`AVAL` covariate value

aml *Acute myelogenous leukemia survival data from the survival package*

Description

Survival in patients with acute myelogenous leukemia.

`time` Survival or censoring time

`status` censoring status

`x` maintenance chemotherapy given or not

Usage

```
aml
```

Format

An object of class `data.frame` with 23 rows and 3 columns.

bscpp *B-Spline Basis for Polynomial Splines*

Description

Computes the B-spline basis matrix for a given polynomial spline.

Usage

```
bscpp(
  x = NA_real_,
  df = NA_integer_,
  knots = NA_real_,
  degree = 3L,
  intercept = 0L,
  boundary_knots = NA_real_,
  warn_outside = 1L
)
```

Arguments

`x` A numeric vector representing the predictor variable.

`df` Degrees of freedom, specifying the number of columns in the basis matrix. If `df` is provided, the function automatically selects `df - degree - intercept` internal knots based on appropriate quantiles of `x`, ignoring any missing values.

`knots` A numeric vector specifying the internal breakpoints that define the spline. If not provided, `df` must be specified.

`degree` An integer specifying the degree of the piecewise polynomial. The default value is `3`, which corresponds to cubic splines.

`intercept` A logical value indicating whether to include an intercept in the basis. The default is `FALSE`.

`boundary_knots` A numeric vector of length 2 specifying the boundary points where the B-spline basis should be anchored. If not supplied, the default is the range of non-missing values in `x`.

`warn_outside` A logical value indicating whether a warning should be issued if any values of `x` fall outside the specified boundary knots.

Value

A matrix with dimensions `c(length(x), df)`. If `df` is provided, the matrix will have `df` columns. Alternatively, if `knots` are supplied, the number of columns will be `length(knots) + degree + intercept`. The matrix contains attributes that correspond to the arguments passed to the `bscpp` function.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
bscpp(women$height, df = 5)
```

`heart` *Stanford heart transplant data from the survival package*

Description

Survival of patients on the waiting list for the Stanford heart transplant program.

`start, stop, event` entry and exit time and status for the time interval

`age` age-48 years

`year` year of acceptance (in years after Nov 1, 1967)

`surgery` prior bypass surgery 1=yes, 0=no

`transplant` received transplant 1=yes, 0=no

`id` patient id

Usage

```
heart
```

Format

An object of class `data.frame` with 172 rows and 8 columns.

`immdef` *Simulated CONCORDE trial data from the rpsftm package*

Description

Patients were randomly assigned to receive treatment immediately or deferred, and those in the deferred arm could cross over and receive treatment. The primary endpoint was time to disease progression.

`id` Patient identification number

`def` Indicator that the participant was assigned to the deferred treatment arm

`imm` Indicator that the participant was assigned to the immediate treatment arm

`censyrs` The censoring time, in years, corresponding to the close of study minus the time of entry for each patient `xo` Indicator that crossover occurred

`xoyrs` The time, in years, from entry to switching, or 0 for patients in the immediate arm

`prog` Indicator of disease progression (1), or censoring (0)

`progyrs` Time, in years, from entry to disease progression or censoring

`entry` The time of entry into the study, measured in years from the date of randomisation

Usage

```
immdef
```

Format

An object of class `data.frame` with 1000 rows and 9 columns.

`ingots` *The binary data from Cox and Snell (1989, pp. 10-11).*

Description

The dataset consits of the number of ingots not ready for rolling and the number of ingots ready for rolling for a number of combinations of heating time and soaking time.

Usage

```
ingots
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 25 rows and 4 columns.

Details

`Heat` The heating time

`Soak` The soaking time

`NotReady` Response indicator, with a value 1 for units not ready for rolling (event) and a value of 0 for units ready for rolling (nonevent)

`Freq` The frequency of occurrence of each combination of `Heat`, `Soak`, and `NotReady`

ipcw *Inverse Probability of Censoring Weights (IPCW) Method for Treatment Switching*

Description

Uses the inverse probability of censoring weights (IPCW) method to obtain the hazard ratio estimate of the Cox model to adjust for treatment switching.

Usage

```
ipcw(
  data,
  id = "id",
  stratum = "",
  tstart = "tstart",
  tstop = "tstop",
  event = "event",
  treat = "treat",
  swtrt = "swtrt",
  swtrt_time = "swtrt_time",
  base_cov = "",
  numerator = "",
  denominator = "",
  logistic_switching_model = FALSE,
  strata_main_effect_only = TRUE,
  firth = FALSE,
  flic = FALSE,
  ns_df = 3,
  stabilized_weights = TRUE,
  trunc = 0,
  trunc_upper_only = TRUE,
  swtrt_control_only = TRUE,
  alpha = 0.05,
  ties = "efron",
  boot = FALSE,
  n_boot = 1000,
  seed = NA
)
```

Arguments

`data` The input data frame that contains the following variables:

- `id`: The id to identify observations belonging to the same subject for counting process data with time-dependent covariates.
- `stratum`: The stratum.

- `tstart`: The starting time of the time interval for counting-process data with time-dependent covariates.
- `tstop`: The stopping time of the time interval for counting-process data with time-dependent covariates.
- `event`: The event indicator, 1=event, 0=no event.
- `treat`: The randomized treatment indicator, 1=treatment, 0=control.
- `swtrt`: The treatment switch indicator, 1=switch, 0=no switch.
- `swtrt_time`: The time from randomization to treatment switch.
- `base_cov`: The baseline covariates (excluding treat) used in the outcome model.
- `numerator`: The baseline covariates (excluding treat) used in the numerator switching model for stabilized weights.
- `denominator`: The baseline (excluding treat) and time-dependent covariates used in the denominator switching model.

`id` The name of the id variable in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`tstart` The name of the tstart variable in the input data.

`tstop` The name of the tstop variable in the input data.

`event` The name of the event variable in the input data.

`treat` The name of the treatment variable in the input data.

`swtrt` The name of the swtrt variable in the input data.

`swtrt_time` The name of the swtrt_time variable in the input data.

`base_cov` The names of baseline covariates (excluding treat) in the input data for the Cox model.

`numerator` The names of baseline covariates (excluding treat) in the input data for the numerator switching model for stabilized weights.

`denominator` The names of baseline (excluding treat) and time-dependent covariates in the input data for the denominator switching model.

`logistic_switching_model`
Whether a pooled logistic regression switching model is used.

`strata_main_effect_only`
Whether to only include the strata main effects in the logistic regression switching model. Defaults to `TRUE`, otherwise all possible strata combinations will be considered in the switching model.

`firth` Whether the Firth's bias reducing penalized likelihood should be used.

`flic` Whether to apply intercept correction to obtain more accurate predicted probabilities.

`ns_df` Degrees of freedom for the natural cubic spline for visit-specific intercepts of the pooled logistic regression model. Defaults to 3 for two internal knots at the 33 and 67 percentiles of the artificial censoring times due to treatment switching.

`stabilized_weights`
Whether to use the stabilized weights. The default is `TRUE`.

`trunc` The truncation fraction of the weight distribution. Defaults to 0 for no truncation in weights.

`trunc_upper_only`
Whether to truncate the weights from the upper end of the weight distribution only. Defaults to `TRUE`, otherwise the weights will be truncated from both the lower and upper ends of the distribution.

`swtrt_control_only`
Whether treatment switching occurred only in the control group. The default is `TRUE`.

`alpha` The significance level to calculate confidence intervals. The default value is 0.05.

`ties` The method for handling ties in the Cox model, either "breslow" or "efron" (default).

`boot` Whether to use bootstrap to obtain the confidence interval for hazard ratio. Defaults to `FALSE`.

`n_boot` The number of bootstrap samples.

`seed` The seed to reproduce the bootstrap results. The default is missing, in which case, the seed from the environment will be used.

Details

We use the following steps to obtain the hazard ratio estimate and confidence interval had there been no treatment switching:

- Exclude observations after treatment switch.
- Set up the crossover and event indicators for the last time interval for each subject.
- For time-dependent covariates Cox switching models, replicate unique event times across treatment arms within each subject.
- Fit the denominator switching model (and the numerator switching model for stabilized weights) to obtain the inverse probability of censoring weights. This can be a Cox model with time-dependent covariates or a pooled logistic regression model. For pooled logistic regression switching model, the probability of remaining uncensored (i.e., not switching) will be calculated by subtracting the predicted probability of switching from 1 and then multiplied over time up to the current time point.
- Fit the weighted Cox model to the censored outcome survival times to obtain the hazard ratio estimate.
- Use either robust sandwich variance or bootstrapping to construct the p-value and confidence interval for the hazard ratio. If bootstrapping is used, the confidence interval and corresponding p-value are calculated based on a t-distribution with `n_boot - 1` degrees of freedom.

Value

A list with the following components:

- `logrank_pvalue`: The two-sided p-value of the log-rank test for the ITT analysis.

- cox_pvalue: The two-sided p-value for treatment effect based on the Cox model applied to counterfactual unswitched survival times. If boot is TRUE, this value represents the bootstrap p-value.
- hr: The estimated hazard ratio from the Cox model.
- hr_CI: The confidence interval for hazard ratio.
- hr_CI_type: The type of confidence interval for hazard ratio, either "Cox model" or "bootstrap".
- data_switch: A list of input data for the switching models by treatment group. The variables include id, stratum, "tstart", "tstop", "cross", denominator, swtrt, and swtrt_time.
- fit_switch: A list of fitted switching models for the denominator and numerator by treatment group.
- data_outcome: The input data for the outcome Cox model including the inverse probability of censoring weights. The variables include id, stratum, "tstart", "tstop", "event", "treated", "unstablized_weight", "stabilized_weight", base_cov, and treat.
- fit_outcome: The fitted outcome Cox model.
- fail: Whether a model fails to converge.
- settings: A list with the following components:
  - logistic_switching_model: Whether a pooled logistic regression switching model is used.
  - strata_main_effect_only: Whether to only include the strata main effects in the logistic regression switching model.
  - firth: Whether the Firth's bias reducing penalized likelihood should be used.
  - flic: Whether to apply intercept correction to obtain more accurate predicted probabilities.
  - ns_df: Degrees of freedom for the natural cubic spline.
  - stabilized_weights: Whether to use the stabilized weights.
  - trunc: The truncation fraction of the weight distribution.
  - trunc_upper_only: Whether to truncate the weights from the upper end of the distribution only.
  - swtrt_control_only Whether treatment switching occurred only in the control group.
  - alpa: The significance level to calculate confidence intervals.
  - ties: The method for handling ties in the Cox model.
  - boot: Whether to use bootstrap to obtain the confidence interval for hazard ratio.
  - n_boot: The number of bootstrap samples.
  - seed: The seed to reproduce the bootstrap results.
- fail_boots: The indicators for failed bootstrap samples if boot is TRUE.
- fail_boots_data: The data for failed bootstrap samples if boot is TRUE.
- hr_boots: The bootstrap hazard ratio estimates if boot is TRUE.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

James M. Robins and Dianne M. Finkelstein. Correcting for noncompliance and dependent censoring in an AIDS clinical trial with inverse probability of censoring weighted (IPCW) log-rank tests. Biometrics. 2000;56(3):779-788.

Examples

```
# Example 1: pooled logistic regression switching model
library(dplyr)

sim1 <- tssim(
  tdxo = 1, coxo = 1, allocation1 = 1, allocation2 = 1,
  p_X_1 = 0.3, p_X_0 = 0.3,
  rate_T = 0.002, beta1 = -0.5, beta2 = 0.3,
  gamma0 = 0.3, gamma1 = -0.9, gamma2 = 0.7, gamma3 = 1.1, gamma4 = -0.8,
  zeta0 = -3.5, zeta1 = 0.5, zeta2 = 0.2, zeta3 = -0.4,
  alpha0 = 0.5, alpha1 = 0.5, alpha2 = 0.4,
  theta1_1 = -0.4, theta1_0 = -0.4, theta2 = 0.2,
  rate_C = 0.0000855, accrualIntensity = 20/30,
  fixedFollowup = FALSE, plannedTime = 1350, days = 30,
  n = 500, NSim = 100, seed = 314159)

fit1 <- ipcw(
  sim1[[1]], id = "id", tstart = "tstart",
  tstop = "tstop", event = "event", treat = "trtrand",
  swtrt = "xo", swtrt_time = "xotime",
  base_cov = "bprog", numerator = "bprog",
  denominator = c("bprog", "L"),
  logistic_switching_model = TRUE, ns_df = 3,
  swtrt_control_only = TRUE, boot = FALSE)

c(fit1$hr, fit1$hr_CI)

# Example 2: time-dependent covariates Cox switching model

fit2 <- ipcw(
  shilong, id = "id", tstart = "tstart", tstop = "tstop",
  event = "event", treat = "bras.f", swtrt = "co",
  swtrt_time = "dco",
  base_cov = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
               "pathway.f"),
  numerator = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
                "pathway.f"),
  denominator = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
                  "pathway.f", "ps", "ttc", "tran"),
  swtrt_control_only = FALSE, boot = FALSE)

c(fit2$hr, fit2$hr_CI)
```

ipe *Iterative Parameter Estimation (IPE) for Treatment Switching*

Description

Obtains the causal parameter estimate from the accelerated failure-time (AFT) model and the hazard ratio estimate from the Cox model to adjust for treatment switching.

Usage

```
ipe(
  data,
  id = "id",
  stratum = "",
  time = "time",
  event = "event",
  treat = "treat",
  rx = "rx",
  censor_time = "censor_time",
  base_cov = "",
  aft_dist = "weibull",
  low_psi = -2,
  hi_psi = 2,
  strata_main_effect_only = 1,
  treat_modifier = 1,
  recensor = TRUE,
  admin_recensor_only = TRUE,
  autoswitch = TRUE,
  root_finding = "brent",
  alpha = 0.05,
  ties = "efron",
  tol = 1e-06,
  boot = FALSE,
  n_boot = 1000,
  seed = NA
)
```

Arguments

data The input data frame that contains the following variables:

- id: The subject id.
- stratum: The stratum.
- time: The survival time for right censored data.
- event: The event indicator, 1=event, 0=no event.
- treat: The randomized treatment indicator, 1=treatment, 0=control.
- rx: The proportion of time on active treatment.

- `censor_time`: The administrative censoring time. It should be provided for all subjects including those who had events.
- `base_cov`: The baseline covariates (excluding treat).

`id` The name of the id variable in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`treat` The name of the treatment variable in the input data.

`rx` The name of the rx variable in the input data.

`censor_time` The name of the censor_time variable in the input data.

`base_cov` The names of baseline covariates (excluding treat) in the input data for the causal AFT model and the outcome Cox model.

`aft_dist` The assumed distribution for time to event for the AFT model. Options include "exponential", "weibull" (default), "loglogistic", and "lognormal".

`low_psi` The lower limit of the causal parameter.

`hi_psi` The upper limit of the causal parameter.

`strata_main_effect_only`
Whether to only include the strata main effects in the AFT model. Defaults to `TRUE`, otherwise all possible strata combinations will be considered in the AFT model.

`treat_modifier` The optional sensitivity parameter for the constant treatment effect assumption.

`recensor` Whether to apply recensoring to counterfactual survival times. Defaults to `TRUE`.

`admin_recensor_only`
Whether to apply recensoring to administrative censoring times only. Defaults to `TRUE`. If `FALSE`, recensoring will be applied to the actual censoring times for dropouts.

`autoswitch` Whether to exclude recensoring for treatment arms with no switching. Defaults to `TRUE`.

`root_finding` Character string specifying the univariate root-finding algorithm to use. Options are `"brent"` (default) for Brent's method, or `"bisection"` for the bisection method.

`alpha` The significance level to calculate confidence intervals.

`ties` The method for handling ties in the Cox model, either "breslow" or "efron" (default).

`tol` The desired accuracy (convergence tolerance) for `psi` for the root finding algorithm.

`boot` Whether to use bootstrap to obtain the confidence interval for hazard ratio. Defaults to `FALSE`, in which case, the confidence interval will be constructed to match the log-rank test p-value.

`n_boot` The number of bootstrap samples.

`seed` The seed to reproduce the bootstrap results. The default is missing, in which case, the seed from the environment will be used.

Details

We use the following steps to obtain the hazard ratio estimate and confidence interval had there been no treatment switching:

- Use IPE to estimate the causal parameter ψ based on the AFT model for the observed survival times for the experimental arm and the counterfactual survival times for the control arm,

$$U_{i,\psi} = T_{C_i} + e^{\psi} T_{E_i}$$

- Fit the Cox proportional hazards model to the observed survival times for the experimental group and the counterfactual survival times for the control group to obtain the hazard ratio estimate.
- Use either the log-rank test p-value for the intention-to-treat (ITT) analysis or bootstrap to construct the confidence interval for hazard ratio. If bootstrapping is used, the confidence interval and corresponding p-value are calculated based on a t-distribution with `n_boot - 1` degrees of freedom.

Value

A list with the following components:

- `psi`: The estimated causal parameter.
- `psi_CI`: The confidence interval for `psi`.
- `psi_CI_type`: The type of confidence interval for `psi`, i.e., "log-rank p-value" or "bootstrap".
- `logrank_pvalue`: The two-sided p-value of the log-rank test for the ITT analysis.
- `cox_pvalue`: The two-sided p-value for treatment effect based on the Cox model applied to counterfactual unswitched survival times. If `boot` is `TRUE`, this value represents the bootstrap p-value.
- `hr`: The estimated hazard ratio from the Cox model.
- `hr_CI`: The confidence interval for hazard ratio.
- `hr_CI_type`: The type of confidence interval for hazard ratio, either "log-rank p-value" or "bootstrap".
- `Sstar`: A data frame containing the counterfactual untreated survival times and event indicators for each treatment group. The variables include `id`, `stratum`, `"t_star"`, `"d_star"`, `"treated"`, `base_cov`, and `treat`.
- `kmstar`: A data frame containing the Kaplan-Meier estimates based on the counterfactual untreated survival times by treatment arm.
- `data_aft`: The input data for the AFT model for estimating `psi`. The variables include `id`, `stratum`, `"t_star"`, `"d_star"`, `"treated"`, `base_cov`, and `treat`.
- `fit_aft`: The fitted AFT model for estimating `psi`.
- `data_outcome`: The input data for the outcome Cox model of counterfactual unswitched survival times. The variables include `id`, `stratum`, `"t_star"`, `"d_star"`, `"treated"`, `base_cov`, and `treat`.
- `fit_outcome`: The fitted outcome Cox model.

- fail: Whether a model fails to converge.
- psimissing: Whether the psi parameter cannot be estimated.
- settings: A list with the following components:
  - aft_dist: The distribution for time to event for the AFT model.
  - strata_main_effect_only: Whether to only include the strata main effects in the AFT model.
  - low_psi: The lower limit of the causal parameter.
  - hi_psi: The upper limit of the causal parameter.
  - treat_modifier: The sensitivity parameter for the constant treatment effect assumption.
  - recensor: Whether to apply recensoring to counterfactual survival times.
  - admin_recensor_only: Whether to apply recensoring to administrative censoring times only.
  - autoswitch: Whether to exclude recensoring for treatment arms with no switching.
  - root_finding: The univariate root-finding algorithm to use.
  - alpha: The significance level to calculate confidence intervals.
  - ties: The method for handling ties in the Cox model.
  - tol: The desired accuracy (convergence tolerance) for psi.
  - boot: Whether to use bootstrap to obtain the confidence interval for hazard ratio.
  - n_boot: The number of bootstrap samples.
  - seed: The seed to reproduce the bootstrap results.
- fail_boots: The indicators for failed bootstrap samples if boot is TRUE.
- fail_boots_data: The data for failed bootstrap samples if boot is TRUE.
- hr_boots: The bootstrap hazard ratio estimates if boot is TRUE.
- psi_boots: The bootstrap psi estimates if boot is TRUE.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Michael Branson and John Whitehead. Estimating a treatment effect in survival studies in which patients switch treatment. Statistics in Medicine. 2002;21(17):2449-2463.

Ian R White. Letter to the Editor: Estimating treatment effects in randomized trials with treatment switching. Statistics in Medicine. 2006;25(9):1619-1622.

Examples

```
library(dplyr)

# Example 1: one-way treatment switching (control to active)

data <- immdef %>% mutate(rx = 1-xoyrs/progyrs)

fit1 <- ipe(
  data, id = "id", time = "progyrs", event = "prog", treat = "imm",
  rx = "rx", censor_time = "censyrs", aft_dist = "weibull",
  boot = FALSE)

c(fit1$hr, fit1$hr_CI)

# Example 2: two-way treatment switching (illustration only)

# the eventual survival time
shilong1 <- shilong %>%
  arrange(bras.f, id, tstop) %>%
  group_by(bras.f, id) %>%
  slice(n()) %>%
  select(-c("ps", "ttc", "tran"))

shilong2 <- shilong1 %>%
  mutate(rx = ifelse(co, ifelse(bras.f == "MTA", dco/ady,
                                1 - dco/ady),
                     ifelse(bras.f == "MTA", 1, 0)))

fit2 <- ipe(
  shilong2, id = "id", time = "tstop", event = "event",
  treat = "bras.f", rx = "rx", censor_time = "dcut",
  base_cov = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
               "pathway.f"),
  aft_dist = "weibull", boot = FALSE)

c(fit2$hr, fit2$hr_CI)
```

kmdiff *Estimate of Milestone Survival Difference*

Description

Obtains the estimate of milestone survival difference between two treatment groups.

Usage

```
kmdiff(
  data,
  rep = "",
  stratum = "",
  treat = "treat",
  time = "time",
  event = "event",
  milestone = NA_real_,
  survDiffH0 = 0,
  conflev = 0.95
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `treat`: The treatment.
- `time`: The possibly right-censored survival time.
- `event`: The event indicator.

`rep` The name of the replication variable in the input data.

`stratum` The name of the stratum variable in the input data.

`treat` The name of the treatment variable in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`milestone` The milestone time at which to calculate the survival probability.

`survDiffH0` The difference in milestone survival probabilities under the null hypothesis. Defaults to 0 for superiority test.

`conflev` The level of the two-sided confidence interval for the difference in milestone survival probabilities. Defaults to 0.95.

Value

A data frame with the following variables:

- `rep`: The replication.
- `milestone`: The milestone time relative to randomization.
- `survDiffH0`: The difference in milestone survival probabilities under the null hypothesis.
- `surv1`: The estimated milestone survival probability for the treatment group.
- `surv2`: The estimated milestone survival probability for the control group.
- `survDiff`: The estimated difference in milestone survival probabilities.
- `vsurv1`: The variance for surv1.
- `vsurv2`: The variance for surv2.
- `vsurvDiff`: The variance for survDiff.
- `survDiffZ`: The Z-statistic value.
- `survDiffPValue`: The one-sided p-value.
- `lower`: The lower bound of confidence interval.
- `upper`: The upper bound of confidence interval.
- `conflev`: The level of confidence interval.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
df <- kmdiff(data = rawdata, rep = "iterationNumber",
             stratum = "stratum", treat = "treatmentGroup",
             time = "timeUnderObservation", event = "event",
             milestone = 12)
head(df)
```

kmest *Kaplan-Meier Estimates of Survival Curve*

Description

Obtains the Kaplan-Meier estimates of the survival curve.

Usage

```
kmest(
  data,
  rep = "",
  stratum = "",
  time = "time",
  event = "event",
  conftype = "log-log",
  conflev = 0.95,
  keep_censor = 0L
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `time`: The possibly right-censored survival time.
- `event`: The event indicator.

`rep` The name(s) of the replication variable(s) in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`conftype` The type of the confidence interval. One of "none", "plain", "log", "log-log" (the default), or "arcsin". The arcsin option bases the intervals on asin(sqrt(survival)).

`conflev` The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.

`keep_censor` Whether to retain the censoring time in the output data frame.

Value

A data frame with the following variables:

- `size`: The number of subjects in the stratum.
- `time`: The event time.
- `nrisk`: The number of subjects at risk.
- `nevent`: The number of subjects having the event.
- `ncensor`: The number of censored subjects.
- `survival`: The Kaplan-Meier estimate of the survival probability.
- `stderr`: The standard error of the estimated survival probability based on the Greendwood formula.
- `lower`: The lower bound of confidence interval if requested.
- `upper`: The upper bound of confidence interval if requested.
- `conflev`: The level of confidence interval if requested.
- `conftype`: The type of confidence interval if requested.
- `stratum`: The stratum.
- `rep`: The replication.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
kmest(data = aml, stratum = "x", time = "time", event = "status")
```

liferegr *Parametric Regression Models for Failure Time Data*

Description

Obtains the parameter estimates from parametric regression models with uncensored, right censored, left censored, or interval censored data.

Usage

```
liferegr(
  data,
  rep = "",
  stratum = "",
  time = "time",
  time2 = "",
  event = "event",
  covariates = "",
  weight = "",
  offset = "",
  id = "",
  dist = "weibull",
  init = NA_real_,
  robust = FALSE,
  plci = FALSE,
  alpha = 0.05,
  maxiter = 50,
  eps = 1e-09
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `time`: The follow-up time for right censored data, or the left end of each interval for interval censored data.
- `time2`: The right end of each interval for interval censored data.
- `event`: The event indicator, 1=event, 0=no event.
- `covariates`: The values of baseline covariates.
- `weight`: The weight for each observation.
- `offset`: The offset for each observation.
- `id`: The optional subject ID to group the score residuals in computing the robust sandwich variance.

`rep` The name(s) of the replication variable(s) in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`time` The name of the time variable or the left end of each interval for interval censored data in the input data.

`time2` The name of the right end of each interval for interval censored data in the input data.

`event` The name of the event variable in the input data for right censored data.

`covariates` The vector of names of baseline covariates in the input data.

`weight` The name of the weight variable in the input data.

`offset` The name of the offset variable in the input data.

`id` The name of the id variable in the input data.

`dist` The assumed distribution for time to event. Options include "exponential", "weibull", "lognormal", and "loglogistic" to be modeled on the log-scale, and "normal" and "logistic" to be modeled on the original scale.

`init` A vector of initial values for the model parameters, including regression coefficients and the log scale parameter. By default, initial values are derived from an intercept-only model. If this approach fails, ordinary least squares (OLS) estimates, ignoring censoring, are used instead.

`robust`	Whether a robust sandwich variance estimate should be computed. In the presence of the id variable, the score residuals will be aggregated for each id when computing the robust sandwich variance estimate.
`plci`	Whether to obtain profile likelihood confidence interval.
`alpha`	The two-sided significance level.
`maxiter`	The maximum number of iterations.
`eps`	The tolerance to declare convergence.

Details

There are two ways to specify the model, one for right censored data through the time and event variables, and the other for interval censored data through the time (lower) and time2 (upper) variables. For the second form, we follow the convention used in SAS PROC LIFEREG:

- If lower is not missing, upper is not missing, and lower is equal to upper, then there is no censoring and the event occurred at time lower.
- If lower is not missing, upper is not missing, and lower < upper, then the event time is censored within the interval (lower, upper).
- If lower is missing, but upper is not missing, then upper will be used as the left censoring value.
- If lower is not missing, but upper is missing, then lower will be used as the right censoring value.
- If lower is not missing, upper is not missing, but lower > upper, or if both lower and upper are missing, then the observation will not be used.

Value

A list with the following components:

- `sumstat`: The data frame of summary statistics of model fit with the following variables:
  - `n`: The number of observations.
  - `nevents`: The number of events.
  - `loglik0`: The log-likelihood under null.
  - `loglik1`: The maximum log-likelihood.
  - `niter`: The number of Newton-Raphson iterations.
  - `dist`: The assumed distribution.
  - `p`: The number of parameters, including the intercept, regression coefficients associated with the covariates, and the log scale parameters for the strata.
  - `nvar`: The number of regression coefficients associated with the covariates (excluding the intercept).
  - `robust`: Whether the robust sandwich variance estimate is requested.
  - `fail`: Whether the model fails to converge.
  - `rep`: The replication.
- `parest`: The data frame of parameter estimates with the following variables:
  - `param`: The name of the covariate for the parameter estimate.

  - beta: The parameter estimate.
  - sebeta: The standard error of parameter estimate.
  - z: The Wald test statistic for the parameter.
  - expbeta: The exponentiated parameter estimate.
  - vbeta: The covariance matrix for parameter estimates.
  - lower: The lower limit of confidence interval.
  - upper: The upper limit of confidence interval.
  - p: The p-value from the chi-square test.
  - method: The method to compute the confidence interval and p-value.
  - sebeta_naive: The naive standard error of parameter estimate if robust variance is requested.
  - vbeta_naive: The naive covariance matrix for parameter estimates if robust variance is requested.
  - rep: The replication.

- p: The number of parameters.
- nvar: The number of columns of the design matrix excluding the intercept.
- param: The parameter names.
- beta: The parameter estimate.
- vbeta: The covariance matrix for parameter estimates.
- vbeta_naive: The naive covariance matrix for parameter estimates.
- terms: The terms object.
- xlevels: A record of the levels of the factors used in fitting.
- data: The input data.
- rep: The name(s) of the replication variable(s).
- stratum: The name(s) of the stratum variable(s).
- time: The name of the time variable.
- time2: The name of the time2 variable.
- event: The name of the event variable.
- covariates: The names of baseline covariates.
- weight: The name of the weight variable.
- offset: The name of the offset variable.
- id: The name of the id variable.
- dist: The assumed distribution for time to event.
- robust: Whether a robust sandwich variance estimate should be computed.
- plci: Whether to obtain profile likelihood confidence interval.
- alpha: The two-sided significance level.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

John D. Kalbfleisch and Ross L. Prentice. The Statistical Analysis of Failure Time Data. Wiley: New York, 1980.

Examples

```
library(dplyr)

# right censored data
(fit1 <- liferegr(
  data = rawdata %>% mutate(treat = 1*(treatmentGroup == 1)),
  rep = "iterationNumber", stratum = "stratum",
  time = "timeUnderObservation", event = "event",
  covariates = "treat", dist = "weibull"))

# tobit regression for left censored data
(fit2 <- liferegr(
  data = tobin %>% mutate(time = ifelse(durable>0, durable, NA)),
  time = "time", time2 = "durable",
  covariates = c("age", "quant"), dist = "normal"))
```

logisregr *Logistic Regression Models for Binary Data*

Description

Obtains the parameter estimates from logistic regression models with binary data.

Usage

```
logisregr(
  data,
  rep = "",
  event = "event",
  covariates = "",
  freq = "",
  weight = "",
  offset = "",
  id = "",
  link = "logit",
  init = NA_real_,
  robust = FALSE,
  firth = FALSE,
  flic = FALSE,
  plci = FALSE,
  alpha = 0.05,
  maxiter = 50,
  eps = 1e-09
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `event`: The event indicator, 1=event, 0=no event.
- `covariates`: The values of baseline covariates.
- `freq`: The frequency for each observation.
- `weight`: The weight for each observation.
- `offset`: The offset for each observation.
- `id`: The optional subject ID to group the score residuals in computing the robust sandwich variance.

`rep` The name(s) of the replication variable(s) in the input data.

`event` The name of the event variable in the input data.

`covariates` The vector of names of baseline covariates in the input data.

`freq` The name of the frequency variable in the input data. The frequencies must be the same for all observations within each cluster as indicated by the id. Thus freq is the cluster frequency.

`weight` The name of the weight variable in the input data.

`offset` The name of the offset variable in the input data.

`id` The name of the id variable in the input data.

`link` The link function linking the response probabilities to the linear predictors. Options include "logit" (default), "probit", and "cloglog" (complementary log-log).

`init` A vector of initial values for the model parameters. By default, initial values are derived from an intercept-only model.

`robust` Whether a robust sandwich variance estimate should be computed. In the presence of the id variable, the score residuals will be aggregated for each id when computing the robust sandwich variance estimate.

`firth` Whether the firth's bias reducing penalized likelihood should be used. The default is `FALSE`.

`flic` Whether to apply intercept correction to obtain more accurate predicted probabilities. The default is `FALSE`.

`plci` Whether to obtain profile likelihood confidence interval.

`alpha` The two-sided significance level.

`maxiter` The maximum number of iterations.

`eps` The tolerance to declare convergence.

Details

Fitting a logistic regression model using Firth's bias reduction method is equivalent to penalization of the log-likelihood by the Jeffreys prior. Firth's penalized log-likelihood is given by

$$l(\beta) + \frac{1}{2}\log(\text{det}(I(\beta)))$$

and the components of the gradient $g(\beta)$ are computed as

$$g(\beta_j) + \frac{1}{2}\text{trace}\left(I(\beta)^{-1}\frac{\partial I(\beta)}{\partial \beta_j}\right)$$

The Hessian matrix is not modified by this penalty.

Firth's method reduces bias in maximum likelihood estimates of coefficients, but it introduces a bias toward one-half in the predicted probabilities.

A straightforward modification to Firth's logistic regression to achieve unbiased average predicted probabilities involves a post hoc adjustment of the intercept. This approach, known as Firth's logistic regression with intercept correction (FLIC), preserves the bias-corrected effect estimates. By excluding the intercept from penalization, it ensures that we don't sacrifice the accuracy of effect estimates to improve the predictions.

Value

A list with the following components:

- `sumstat`: The data frame of summary statistics of model fit with the following variables:
  - `n`: The number of subjects.
  - `nevents`: The number of events.
  - `loglik0`: The (penalized) log-likelihood under null.
  - `loglik1`: The maximum (penalized) log-likelihood.
  - `niter`: The number of Newton-Raphson iterations.
  - `p`: The number of parameters, including the intercept, and regression coefficients associated with the covariates.
  - `link`: The link function.
  - `robust`: Whether a robust sandwich variance estimate should be computed.
  - `firth`: Whether the firth's penalized likelihood is used.
  - `flic`: Whether to apply intercept correction.
  - `fail`: Whether the model fails to converge.
  - `loglik0_unpenalized`: The unpenalized log-likelihood under null.
  - `loglik1_unpenalized`: The maximum unpenalized log-likelihood.
  - `rep`: The replication.
- `parest`: The data frame of parameter estimates with the following variables:
  - `param`: The name of the covariate for the parameter estimate.
  - `beta`: The parameter estimate.
  - `sebeta`: The standard error of parameter estimate.
  - `z`: The Wald test statistic for the parameter.

  - `expbeta`: The exponentiated parameter estimate.
  - `vbeta`: The covariance matrix for parameter estimates.
  - `lower`: The lower limit of confidence interval.
  - `upper`: The upper limit of confidence interval.
  - `p`: The p-value from the chi-square test.
  - `method`: The method to compute the confidence interval and p-value.
  - `sebeta_naive`: The naive standard error of parameter estimate.
  - `vbeta_naive`: The naive covariance matrix of parameter estimates.
  - `rep`: The replication.
- `fitted`: The data frame with the following variables:
  - `linear_predictors`: The linear fit on the link function scale.
  - `fitted_values`: The fitted probabilities of having an event, obtained by transforming the linear predictors by the inverse of the link function.
  - `rep`: The replication.
- `p`: The number of parameters.
- `link`: The link function.
- `param`: The parameter names.
- `beta`: The parameter estimate.
- `vbeta`: The covariance matrix for parameter estimates.
- `vbeta_naive`: The naive covariance matrix for parameter estimates.
- `linear_predictors`: The linear fit on the link function scale.
- `fitted_values`: The fitted probabilities of having an event.
- `terms`: The terms object.
- `xlevels`: A record of the levels of the factors used in fitting.
- `data`: The input data.
- `rep`: The name(s) of the replication variable(s).
- `event`: The name of the event variable.
- `covariates`: The names of baseline covariates.
- `freq`: The name of the freq variable.
- `weight`: The name of the weight variable.
- `offset`: The name of the offset variable.
- `id`: The name of the id variable.
- `robust`: Whether a robust sandwich variance estimate should be computed.
- `firth`: Whether to use the firth's bias reducing penalized likelihood.
- `flic`: Whether to apply intercept correction.
- `plci`: Whether to obtain profile likelihood confidence interval.
- `alpha`: The two-sided significance level.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

David Firth. Bias Reduction of Maximum Likelihood Estimates. Biometrika 1993; 80:27–38.

Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. Statistics in Medicine 2002;21:2409–2419.

Rainer Puhr, Georg Heinze, Mariana Nold, Lara Lusa, and Angelika Geroldinger. Firth's logistic regression with rare events: accurate effect estimates and predictions? Statistics in Medicine 2017; 36:2302-2317.

Examples

```
(fit1 <- logisregr(
  ingots, event = "NotReady", covariates = "Heat*Soak", freq = "Freq"))
```

lrtest *Log-Rank Test of Survival Curve Difference*

Description

Obtains the log-rank test using the Fleming-Harrington family of weights.

Usage

```
lrtest(
  data,
  rep = "",
  stratum = "",
  treat = "treat",
  time = "time",
  event = "event",
  rho1 = 0,
  rho2 = 0
)
```

Arguments

data The input data frame that contains the following variables:

- rep: The replication for by-group processing.
- stratum: The stratum.
- treat: The treatment.
- time: The possibly right-censored survival time.
- event: The event indicator.

`rep` The name of the replication variable in the input data.

`stratum` The name of the stratum variable in the input data.

`treat` The name of the treatment variable in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`rho1` The first parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

`rho2` The second parameter of the Fleming-Harrington family of weighted log-rank test. Defaults to 0 for conventional log-rank test.

Value

A data frame with the following variables:

- `uscore`: The numerator of the log-rank test statistic.
- `vscore`: The variance of the log-rank score test statistic.
- `logRankZ`: The Z-statistic value.
- `logRankPValue`: The one-sided p-value.
- `rho1`: The first parameter of the Fleming-Harrington weights.
- `rho2`: The second parameter of the Fleming-Harrington weights.
- `rep`: The replication.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
df <- lrtest(data = rawdata, rep = "iterationNumber",
             stratum = "stratum", treat = "treatmentGroup",
             time = "timeUnderObservation", event = "event",
             rho1 = 0.5, rho2 = 0)
head(df)
```

`msm` *Marginal Structural Model (MSM) Method for Treatment Switching*

Description

Uses the marginal structural model (MSM) method to obtain the hazard ratio estimate of the Cox model to adjust for treatment switching.

Usage

```
msm(
  data,
  id = "id",
  stratum = "",
  tstart = "tstart",
  tstop = "tstop",
  event = "event",
  treat = "treat",
  swtrt = "swtrt",
  swtrt_time = "swtrt_time",
  base_cov = "",
  numerator = "",
  denominator = "",
  strata_main_effect_only = TRUE,
  firth = FALSE,
  flic = FALSE,
  ns_df = 3,
  stabilized_weights = TRUE,
  trunc = 0,
  trunc_upper_only = TRUE,
  swtrt_control_only = TRUE,
  treat_alt_interaction = TRUE,
  alpha = 0.05,
  ties = "efron",
  boot = FALSE,
  n_boot = 1000,
  seed = NA
)
```

Arguments

`data` The input data frame that contains the following variables:

- `id`: The id to identify observations belonging to the same subject for counting process data with time-dependent covariates.
- `stratum`: The stratum.
- `tstart`: The starting time of the time interval for counting-process data with time-dependent covariates.
- `tstop`: The stopping time of the time interval for counting-process data with time-dependent covariates.
- `event`: The event indicator, 1=event, 0=no event.
- `treat`: The randomized treatment indicator, 1=treatment, 0=control.
- `swtrt`: The treatment switch indicator, 1=switch, 0=no switch.
- `swtrt_time`: The time from randomization to treatment switch.
- `base_cov`: The baseline covariates (excluding treat) used in the outcome model.

- `numerator`: The baseline covariates (excluding treat) used in the numerator switching model for stabilized weights.
- `denominator`: The baseline (excluding treat) and time-dependent covariates used in the denominator switching model.

`id` The name of the id variable in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`tstart` The name of the tstart variable in the input data.

`tstop` The name of the tstop variable in the input data.

`event` The name of the event variable in the input data.

`treat` The name of the treatment variable in the input data.

`swtrt` The name of the swtrt variable in the input data.

`swtrt_time` The name of the swtrt_time variable in the input data.

`base_cov` The names of baseline covariates (excluding treat) in the input data for the Cox model.

`numerator` The names of baseline covariates (excluding treat) in the input data for the numerator switching model for stabilized weights.

`denominator` The names of baseline (excluding treat) and time-dependent covariates in the input data for the denominator switching model.

`strata_main_effect_only`
Whether to only include the strata main effects in the logistic regression switching model. Defaults to `TRUE`, otherwise all possible strata combinations will be considered in the switching model.

`firth` Whether the Firth's bias reducing penalized likelihood should be used.

`flic` Whether to apply intercept correction to obtain more accurate predicted probabilities.

`ns_df` Degrees of freedom for the natural cubic spline for visit-specific intercepts of the pooled logistic regression model. Defaults to 3 for two internal knots at the 33 and 67 percentiles of the artificial censoring times due to treatment switching.

`stabilized_weights`
Whether to use the stabilized weights. The default is `TRUE`.

`trunc` The truncation fraction of the weight distribution. Defaults to 0 for no truncation in weights.

`trunc_upper_only`
Whether to truncate the weights from the upper end of the weight distribution only. Defaults to `TRUE`, otherwise the weights will be truncated from both the lower and upper ends of the distribution.

`swtrt_control_only`
Whether treatment switching occurred only in the control group. The default is `TRUE`.

`treat_alt_interaction`
Whether to include an interaction between randomized and alternative treatment in the outcome model when both randomized arms can switch to alternative treatment.

`alpha` The significance level to calculate confidence intervals. The default value is 0.05.

`ties` The method for handling ties in the Cox model, either "breslow" or "efron" (default).

`boot` Whether to use bootstrap to obtain the confidence interval for hazard ratio. Defaults to `FALSE`.

`n_boot` The number of bootstrap samples.

`seed` The seed to reproduce the bootstrap results. The default is missing, in which case, the seed from the environment will be used.

Details

We use the following steps to obtain the hazard ratio estimate and confidence interval had there been no treatment switching:

- Exclude observations after treatment switch for the switch model.
- Set up the crossover indicators for the last time interval for each subject.
- Fit the denominator switching model (and the numerator switching model for stabilized weights) to obtain the inverse probability of censoring weights using a pooled logistic regression model. The probability of remaining uncensored (i.e., not switching) will be calculated by subtracting the predicted probability of switching from 1. The probabilities of remaining unswitched will be multiplied over time before treatment switching, at which time point, it will be multiplied by the probability of treatment switching. The inverse probability of treatment weighting will not change after treatment switching.
- Fit the weighted Cox model to the censored outcome survival times to obtain the hazard ratio estimate.
- Use either robust sandwich variance or bootstrapping to construct the p-value and confidence interval for the hazard ratio. If bootstrapping is used, the confidence interval and corresponding p-value are calculated based on a t-distribution with `n_boot - 1` degrees of freedom.

Value

A list with the following components:

- `logrank_pvalue`: The two-sided p-value of the log-rank test for the ITT analysis.
- `cox_pvalue`: The two-sided p-value for treatment effect based on the Cox model applied to counterfactual unswitched survival times. If `boot` is `TRUE`, this value represents the bootstrap p-value.
- `hr`: The estimated hazard ratio from the Cox model.
- `hr_CI`: The confidence interval for hazard ratio.
- `hr_CI_type`: The type of confidence interval for hazard ratio, either "Cox model" or "bootstrap".
- `data_switch`: A list of input data for the switching models by treatment group. The variables include `id`, `stratum`, `"tstart"`, `"tstop"`, `"cross"`, `denominator`, `swtrt`, and `swtrt_time`.
- `fit_switch`: A list of fitted switching models for the denominator and numerator by treatment group.

- data_outcome: The input data for the outcome Cox model including the inverse probability of censoring weights. The variables include id, stratum, "tstart", "tstop", "event", "treated", "unstablized_weight", "stabilized_weight", base_cov, and treat.
- fit_outcome: The fitted outcome Cox model.
- fail: Whether a model fails to converge.
- settings: A list with the following components:
  - strata_main_effect_only: Whether to only include the strata main effects in the logistic regression switching model.
  - firth: Whether the Firth's bias reducing penalized likelihood should be used.
  - flic: Whether to apply intercept correction to obtain more accurate predicted probabilities.
  - ns_df: Degrees of freedom for the natural cubic spline.
  - stabilized_weights: Whether to use the stabilized weights.
  - trunc: The truncation fraction of the weight distribution.
  - trunc_upper_only: Whether to truncate the weights from the upper end of the distribution only.
  - swtrt_control_only Whether treatment switching occurred only in the control group.
  - treat_alt_interaction Whether to include an interaction between randomized and alternative treatment in the outcome model.
  - alpa: The significance level to calculate confidence intervals.
  - ties: The method for handling ties in the Cox model.
  - boot: Whether to use bootstrap to obtain the confidence interval for hazard ratio.
  - n_boot: The number of bootstrap samples.
  - seed: The seed to reproduce the bootstrap results.
- fail_boots: The indicators for failed bootstrap samples if boot is TRUE.
- fail_boots_data: The data for failed bootstrap samples if boot is TRUE.
- hr_boots: The bootstrap hazard ratio estimates if boot is TRUE.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

James M. Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. Epidemiology. 2000;11(5):550-560.

Miguel Angel Hernan, Babette Brumback, and James M. Robins. Marginal structural modesl to estimate the causual effect of zidovudine on the survival of HIV-positive men. Epidemiology. 2000;11(5):561-570.

Jing Xu, Guohui Liu, and Bingxia Wang. Bias and Type I error control in correcting treatment effect for treatment switching using marginal structural models in Phase III oncology trials. Journal of Biopharmaceutical Statistics. 2022;32(6):897-914.

Examples

```
library(dplyr)

sim1 <- tssim(
  tdxo = 1, coxo = 1, allocation1 = 1, allocation2 = 1,
  p_X_1 = 0.3, p_X_0 = 0.3,
  rate_T = 0.002, beta1 = -0.5, beta2 = 0.3,
  gamma0 = 0.3, gamma1 = -0.9, gamma2 = 0.7, gamma3 = 1.1, gamma4 = -0.8,
  zeta0 = -3.5, zeta1 = 0.5, zeta2 = 0.2, zeta3 = -0.4,
  alpha0 = 0.5, alpha1 = 0.5, alpha2 = 0.4,
  theta1_1 = -0.4, theta1_0 = -0.4, theta2 = 0.2,
  rate_C = 0.0000855, accrualIntensity = 20/30,
  fixedFollowup = FALSE, plannedTime = 1350, days = 30,
  n = 500, NSim = 100, seed = 314159)

fit1 <- msm(
  sim1[[1]], id = "id", tstart = "tstart",
  tstop = "tstop", event = "event", treat = "trtrand",
  swtrt = "xo", swtrt_time = "xotime",
  base_cov = "bprog", numerator = "bprog",
  denominator = c("bprog", "L"),
  ns_df = 3, swtrt_control_only = TRUE, boot = FALSE)

c(fit1$hr, fit1$hr_CI)
```

nscpp *Natural Cubic Spline Basis*

Description

Computes the B-spline basis matrix for a natural cubic spline.

Usage

```
nscpp(
  x = NA_real_,
  df = NA_integer_,
  knots = NA_real_,
  intercept = 0L,
  boundary_knots = NA_real_
)
```

Arguments

x	A numeric vector representing the predictor variable. Missing values are allowed.

df	Degrees of freedom, specifying the number of columns in the basis matrix. If df is provided, the function selects df - 1 - intercept internal knots based on appropriate quantiles of x, ignoring any missing values.
knots	A numeric vector specifying the internal breakpoints that define the spline. If provided, the number of degrees of freedom will be determined by the length of knots.
intercept	A logical value indicating whether to include an intercept in the basis. The default is FALSE.
boundary_knots	A numeric vector of length 2 specifying the boundary points where the natural boundary conditions are applied and the B-spline basis is anchored. If not supplied, the default is the range of non-missing values in x.

Value

A matrix with dimensions c(length(x), df), where df is either provided directly or computed as length(knots) + 1 + intercept when knots are supplied. The matrix contains attributes that correspond to the arguments passed to the nscpp function.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
nscpp(women$height, df = 5)
```

---

phregr *Proportional Hazards Regression Models*

Description

Obtains the hazard ratio estimates from the proportional hazards regression model with right censored or counting process data.

Usage

```
phregr(
  data,
  rep = "",
  stratum = "",
  time = "time",
  time2 = "",
  event = "event",
  covariates = "",
  weight = "",
  offset = "",
  id = "",
  ties = "efron",
  init = NA_real_,
  robust = FALSE,
  est_basehaz = TRUE,
  est_resid = TRUE,
  firth = FALSE,
  plci = FALSE,
  alpha = 0.05,
  maxiter = 50,
  eps = 1e-09
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `time`: The follow-up time for right censored data, or the left end of each interval for counting process data.
- `time2`: The right end of each interval for counting process data. Intervals are assumed to be open on the left and closed on the right, and event indicates whether an event occurred at the right end of each interval.
- `event`: The event indicator, 1=event, 0=no event.
- `covariates`: The values of baseline covariates (and time-dependent covariates in each interval for counting process data).
- `weight`: The weight for each observation.
- `offset`: The offset for each observation.
- `id`: The optional subject ID for counting process data with time-dependent covariates.

`rep` The name(s) of the replication variable(s) in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`time` The name of the time variable or the left end of each interval for counting process data in the input data.

`time2` The name of the right end of each interval for counting process data in the input data.

`event` The name of the event variable in the input data.

`covariates` The vector of names of baseline and time-dependent covariates in the input data.

`weight` The name of the weight variable in the input data.

`offset` The name of the offset variable in the input data.

`id` The name of the id variable in the input data.

`ties` The method for handling ties, either "breslow" or "efron" (default).

`init` The vector of initial values. Defaults to zero for all variables.

`robust` Whether a robust sandwich variance estimate should be computed. In the presence of the id variable, the score residuals will be aggregated for each id when computing the robust sandwich variance estimate.

`est_basehaz` Whether to estimate the baseline hazards. Defaults to `TRUE`.

`est_resid` Whether to estimate the martingale residuals. Defaults to `TRUE`.

`firth` Whether to use Firth's penalized likelihood method. Defaults to `FALSE`.

`plci` Whether to obtain profile likelihood confidence interval.

`alpha` The two-sided significance level.

`maxiter` The maximum number of iterations.

`eps` The tolerance to declare convergence.

Value

A list with the following components:

- `sumstat`: The data frame of summary statistics of model fit with the following variables:
  - `n`: The number of observations.
  - `nevents`: The number of events.
  - `loglik0`: The (penalized) log-likelihood under null.
  - `loglik1`: The maximum (penalized) log-likelihood.
  - `scoretest`: The score test statistic.
  - `niter`: The number of Newton-Raphson iterations.
  - `ties`: The method for handling ties, either "breslow" or "efron".
  - `p`: The number of columns of the Cox model design matrix.
  - `robust`: Whether to use the robust variance estimate.
  - `firth`: Whether to use Firth's penalized likelihood method.
  - `fail`: Whether the model fails to converge.
  - `loglik0_unpenalized`: The unpenalized log-likelihood under null.
  - `loglik1_unpenalized`: The maximum unpenalized log-likelihood.
  - `rep`: The replication.
- `parest`: The data frame of parameter estimates with the following variables:
  - `param`: The name of the covariate for the parameter estimate.
  - `beta`: The log hazard ratio estimate.
  - `sebeta`: The standard error of log hazard ratio estimate.
  - `z`: The Wald test statistic for log hazard ratio.
  - `expbeta`: The hazard ratio estimate.
  - `vbeta`: The covariance matrix for parameter estimates.
  - `lower`: The lower limit of confidence interval.
  - `upper`: The upper limit of confidence interval.
  - `p`: The p-value from the chi-square test.
  - `method`: The method to compute the confidence interval and p-value.
  - `sebeta_naive`: The naive standard error of log hazard ratio estimate if robust variance is requested.

  - vbeta_naive: The naive covariance matrix for parameter estimates if robust variance is requested.
  - rep: The replication.
- basehaz: The data frame of baseline hazards with the following variables (if est_basehaz is TRUE):
  - time: The observed event time.
  - nrisk: The number of patients at risk at the time point.
  - nevent: The number of events at the time point.
  - haz: The baseline hazard at the time point.
  - varhaz: The variance of the baseline hazard at the time point assuming the parameter beta is known.
  - gradhaz: The gradient of the baseline hazard with respect to beta at the time point (in the presence of covariates).
  - stratum: The stratum.
  - rep: The replication.
- residuals: The martingale residuals.
- p: The number of parameters.
- param: The parameter names.
- beta: The parameter estimate.
- vbeta: The covariance matrix for parameter estimates.
- vbeta_naive: The naive covariance matrix for parameter estimates.
- terms: The terms object.
- xlevels: A record of the levels of the factors used in fitting.
- data: The input data.
- rep: The name(s) of the replication variable(s).
- stratum: The name(s) of the stratum variable(s).
- time: The name of the time varaible.
- time2: The name of the time2 variable.
- event: The name of the event variable.
- covariates: The names of baseline covariates.
- weight: The name of the weight variable.
- offset: The name of the offset variable.
- id: The name of the id variable.
- ties: The method for handling ties.
- robust: Whether a robust sandwich variance estimate should be computed.
- est_basehaz: Whether to estimate the baseline hazards.
- est_resid: Whether to estimate the martingale residuals.
- firth: Whether to use Firth’s penalized likelihood method.
- plci: Whether to obtain profile likelihood confidence interval.
- alpha: The two-sided significance level.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Per K. Anderson and Richard D. Gill. Cox's regression model for counting processes, a large sample study. Annals of Statistics 1982; 10:1100-1120.

Terry M. Therneau and Patricia M. Grambsch. Modeling Survival Data: Extending the Cox Model. Springer-Verlag, 2000.

Examples

```
library(dplyr)

# Example 1 with right-censored data
(fit1 <- phregr(
  data = rawdata %>% mutate(treat = 1*(treatmentGroup == 1)),
  rep = "iterationNumber", stratum = "stratum",
  time = "timeUnderObservation", event = "event",
  covariates = "treat", est_basehaz = FALSE, est_resid = FALSE))

# Example 2 with counting process data and robust variance estimate
(fit2 <- phregr(
  data = heart %>% mutate(rx = as.numeric(transplant) - 1),
  time = "start", time2 = "stop", event = "event",
  covariates = c("rx", "age"), id = "id",
  robust = TRUE, est_basehaz = TRUE, est_resid = TRUE))
```

preptdc *Prepare Survival Data With Time-Dependent Covariates*

Description

This function prepares a counting-process style survival dataset for analyses with time-dependent covariates. It merges baseline and longitudinal data, fills in missing covariate values using last-observation-carried-forward (LOCF), restricts to time points where covariates change (optional), and constructs `tstart`, `tstop`, and `event` variables suitable for use in survival models.

Usage

```
preptdc(
  adsl,
  adtdc,
  id = "SUBJID",
  randdt = "RANDDT",
  trtsdt = "TRTSDT",
  pddt = "PDDT",
  xodt = "XODT",
  osdt = "OSDT",
  died = "DIED",
  dcutdt = "DCUTDT",
  adt = "ADT",
  paramcd = "PARAMCD",
  aval = "AVAL",
  nodup = TRUE,
  offset = TRUE
)
```

Arguments

adsl	A data set containing baseline subject-level information. It should include, at a minimum, subject ID (id), randomization date (randdt), treatment start date (trtsdt), survival outcome (osdt, died), progression date (pddt), treatment switch date (xodt), and data cut-off date (dcutdt).
adtdc	A data set containing longitudinal time-dependent covariate data, with subject ID (id), parameter code (paramcd), analysis date (adt), and covariate value (aval).
id	Character string specifying the column name for subject ID.
randdt	Character string specifying the column name for randomization date.
trtsdt	Character string specifying the column name for treatment start date.
pddt	Character string specifying the column name for progression date.
xodt	Character string specifying the column name for treatment crossover/switch date.
osdt	Character string specifying the column name for overall survival date (death date or last known alive date).
died	Character string specifying the column name for death indicator (0 = alive/censored, 1 = died).
dcutdt	Character string specifying the column name for data cut-off date.
adt	Character string specifying the column name for analysis date in the time-dependent covariate dataset.
paramcd	Character string specifying the column name for parameter code (identifying different covariates).
aval	Character string specifying the column name for analysis value (covariate values).
nodup	Logical; if TRUE (default), only rows where at least one covariate changes compared to the previous row (within each subject) are retained, along with the first row per subject (baseline).
offset	Logical; if TRUE (default), add 1-day offset when computing analysis day variables (ady, osdy, etc.).

Details

The function performs the following steps:

1. Merge adsl and adtdc to obtain randomization date and treatment start date.
2. Define adt2 as adt if adt > trtsdt, and randdt if adt <= trtsdt (i.e., baseline time point). This ensures that the baseline covariate value is the last non-missing value at or before the treatment start date. Post-baseline covariate values are anchored at their actual analysis dates.
3. Keep the last record per subject, adt2, and paramcd.
4. Construct a complete skeleton so all covariates are present for each subject and time point.
5. Fill missing covariate values using LOCF.
6. Pivot to wide format with one row per subject and time point.
7. Optionally drop rows without covariate changes (nodup = TRUE).
8. Merge survival outcomes from adsl.
9. Compute time-to-event variables (ady, osdy, etc.), as well as counting-process style variables tstart, tstop, and event.

Value

A data set with one row per subject and time interval, including:

- tstart, tstop — interval start and stop times (days from randomization).
- event — event indicator (0/1).
- Covariates expanded to wide format.
- Auxiliary variables such as progression indicator (pd), treatment switch indicator (swtrt), and administrative censoring time.

Examples

```
surv_data <- preptdc(adsl, adtdc, nodup = TRUE)
head(surv_data)
```

---

qrcpp *QR Decomposition of a Matrix*

Description

Computes the QR decomposition of a matrix.

Usage

```
qrcpp(X, tol = 1e-12)
```

Arguments

`X` A numeric matrix whose QR decomposition is to be computed.

`tol` The tolerance for detecting linear dependencies in the columns of `X`.

Details

This function performs Householder QR with column pivoting: Given an m-by-n matrix A with $m \geq n$, the following algorithm computes $r = \text{rank}(A)$ and the factorization $Q^T AP$ equal to

$$\begin{array}{cccccc} | & R_{11} & R_{12} & | & r \\ | & 0 & 0 & | & m-r \\ & r & n-r & & \end{array}$$

with $Q = H_1 \cdots H_r$ and $P = P_1 \cdots P_r$. The upper triangular part of A is overwritten by the upper triangular part of R and components $(j+1) : m$ of the jth Householder vector are stored in $A((j+1) : m, j)$. The permutation P is encoded in an integer vector `pivot`.

Value

A list with the following components:

- `qr`: A matrix with the same dimensions as `X`. The upper triangle contains the `R` of the decomposition and the lower triangle contains Householder vectors (stored in compact form).
- `rank`: The rank of `X` as computed by the decomposition.
- `pivot`: The column permutation for the pivoting strategy used during the decomposition.
- `Q`: The complete m-by-m orthogonal matrix Q.
- `R`: The complete m-by-n upper triangular matrix R.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

References

Gene N. Golub and Charles F. Van Loan. Matrix Computations, second edition. Baltimore, Maryland: The John Hopkins University Press, 1989, p.235.

Examples

```
hilbert <- function(n) { i <- 1:n; 1 / outer(i - 1, i, `+`) }
h9 <- hilbert(9)
qrcpp(h9)
```

rawdata *A simulated time-to-event data set with 10 replications*

Description

A simulated data set with stratification and delayed treatment effect:

`iterationNumber` The iteration number

`arrivalTime` The enrollment time for the subject

`stratum` The stratum for the subject

`treatmentGroup` The treatment group for the subject

`timeUnderObservation` The time under observation since randomization

`event` Whether the subject experienced the event

`dropoutEvent` Whether the subject dropped out

Usage

```
rawdata
```

Format

An object of class `data.frame` with 4910 rows and 7 columns.

recensor_sim_rpsftm *Simulation Study to Evaluate Recensoring Rules in RPSFTM*

Description

Simulates datasets to evaluate the performance of various recensoring strategies under the Rank Preserving Structural Failure Time Model (RPSFTM) for handling treatment switching in survival analysis.

Usage

```
recensor_sim_rpsftm(
  nsim = NA_integer_,
  n = NA_integer_,
  shape = NA_real_,
  scale = NA_real_,
  gamma = NA_real_,
  tfmin = NA_real_,
  tfmax = NA_real_,
  psi = NA_real_,
  omega = NA_real_,
  pswitch = NA_real_,
  a = NA_real_,
  b = NA_real_,
  low_psi = -1,
  hi_psi = 1,
  treat_modifier = 1,
  recensor_type = 1L,
  admin_recensor_only = 1L,
  autoswitch = 1L,
  alpha = 0.05,
  ties = "efron",
  tol = 1e-06,
  boot = 1L,
  n_boot = 1000L,
  seed = NA_integer_
)
```

Arguments

`nsim`	Number of simulated datasets.
`n`	Number of subjects per simulation.
`shape`	Shape parameter of the Weibull distribution for time to death.
`scale`	Scale parameter of the Weibull distribution for time to death in the control group.
`gamma`	Rate parameter of the exponential distribution for random dropouts in the control group.
`tfmin`	Minimum planned follow-up time (in days).
`tfmax`	Maximum planned follow-up time (in days).
`psi`	Log time ratio of death time for control vs experimental treatment.
`omega`	Log time ratio of dropout time for control vs experimental treatment.
`pswitch`	Probability of treatment switching at disease progression.
`a`	Shape parameter 1 of the Beta distribution for time to disease progression as a fraction of time to death.
`b`	Shape parameter 2 of the Beta distribution for time to disease progression.
`low_psi`	Lower bound for the search interval of the causal parameter ψ.
`hi_psi`	Upper bound for the search interval of the causal parameter ψ.
`treat_modifier`	Sensitivity parameter modifying the constant treatment effect assumption.
`recensor_type`	Type of recensoring to apply: • 0: No recensoring • 1: Recensor all control-arm subjects • 2: Recensor only switchers in the control arm • 3: Recensor only control-arm switchers whose counterfactual survival exceeds the planned follow-up time

admin_recensor_only
: Logical. If TRUE, recensoring is applied only to administrative censoring times. If FALSE, it is also applied to dropout times.

autoswitch
: Logical. If TRUE, disables recensoring in arms without any treatment switching.

alpha
: Significance level for confidence interval calculation (default is 0.05).

ties
: Method for handling tied event times in the Cox model. Options are "efron" (default) or "breslow".

tol
: Convergence tolerance for root-finding in estimation of ψ.

boot
: Logical. If TRUE, bootstrap is used to estimate the confidence interval for the hazard ratio. If FALSE, the confidence interval is matched to the log-rank p-value.

n_boot
: Number of bootstrap samples, used only if boot = TRUE.

seed
: Optional. Random seed for reproducibility. If not provided, the global seed is used.

Value

A data frame summarizing the simulation results, including:

- recensor_type, admin_recensor_only: Settings used in the simulation.
- Event rates: p_event_1, p_dropout_1, p_admin_censor_1, p_event_0, p_dropout_0, p_admin_censor_0.
- Progression and switching: p_pd_0, p_swtrt_0, p_recensored_0.
- Causal parameter (ψ) estimates: psi, psi_est, psi_bias, psi_se, psi_mse.
- Log hazard ratio estimates: loghr, loghr_est, loghr_se, loghr_mse.
- Hazard ratio metrics: hr, hr_est (geometric mean), hr_pctbias (percent bias).
- Standard errors of log hazard ratio: loghr_se_cox, loghr_se_lr, loghr_se_boot.
- Coverage probabilities: hr_ci_cover_cox, hr_ci_cover_lr, hr_ci_cover_boot.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
result <- recensor_sim_rpsftm(
  nsim = 10, n = 400, shape = 1.5, scale = exp(6.3169),
  gamma = 0.001, tfmin = 407.5, tfmax = 407.5,
  psi = log(0.5) / 1.5, omega = log(1), pswitch = 0.7,
  a = 2, b = 4, low_psi = -5, hi_psi = 5,
  treat_modifier = 1, recensor_type = 1,
  admin_recensor_only = TRUE, autoswitch = TRUE,
  alpha = 0.05, tol = 1e-6, boot = TRUE,
  n_boot = 10, seed = 314159)
```

residuals_liferegr *Residuals for Parametric Regression Models for Failure Time Data*

Description

Obtains the response, martingale, deviance, dfbeta, and likelihood displacement residuals for a parametric regression model for failure time data.

Usage

```
residuals_liferegr(
  object,
  type = c("response", "martingale", "deviance", "dfbeta", "dfbetas", "working",
    "ldcase", "ldresp", "ldshape", "matrix"),
  collapse = FALSE,
  weighted = (type %in% c("dfbeta", "dfbetas"))
)
```

Arguments

`object`	The output from the `phregr` call.
`type`	The type of residuals desired, with options including `"response"`, `"martingale"`, `"deviance"`, `"dfbeta"`, `"dfbetas"`, `"working"`, `"ldcase"`, `"ldresp"`, `"ldshape"`, and `"matrix"`.
`collapse`	Whether to collapse the residuals by `id`.
`weighted`	Whether to compute weighted residuals.

Details

The algorithms follow the `residuals.survreg` function in the `survival` package, except for martingale residuals, which are defined only for event or right-censored data for exponential, weibull, lognormal, and loglogistic distributions.

Value

Either a vector or a matrix of residuals, depending on the specified type:

- `response` residuals are on the scale of the original data.
- `martingale` residuals are event indicators minus the cumulative hazards for event or right-censored data.
- `working` residuals are on the scale of the linear predictor.
- `deviance` residuals are on the log-likelihood scale.
- `dfbeta` residuals are returned as a matrix, where the i-th row represents the approximate change in the model coefficients resulting from the inclusion of subject i.
- `dfbetas` residuals are similar to `dfbeta` residuals, but each column is scaled by the standard deviation of the corresponding coefficient.

- matrix residuals are a matrix of derivatives of the log-likelihood function. Let L be the log-likelihood, p be the linear predictor ($X\beta$), and s be $log(\sigma)$. Then the resulting matrix contains six columns: L, $\partial L/\partial p$, $\partial^2 L/\partial p^2$, $\partial L/\partial s$, $\partial^2 L/\partial s^2$, and $\partial L^2/\partial p \partial s$.
- ldcase residulas are likelihood displacement for case weight perturbation.
- ldresp residuals are likelihood displacement for response value perturbation.
- ldshape residuals are likelihood displacement related to the shape parameter.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Escobar, L. A. and Meeker, W. Q. Assessing influence in regression analysis with censored data. Biometrics 1992; 48:507-528.

Examples

```
library(dplyr)

fit1 <- liferegr(
  data = tobin %>% mutate(time = ifelse(durable>0, durable, NA)),
  time = "time", time2 = "durable",
  covariates = c("age", "quant"), dist = "normal")

resid <- residuals_liferegr(fit1, type = "response")
head(resid)
```

residuals_phregr *Residuals for Proportional Hazards Regression Models*

Description

Obtains the martingale, deviance, score, or Schoenfeld residuals for a proportional hazards regression model.

Usage

```
residuals_phregr(
  object,
  type = c("martingale", "deviance", "score", "schoenfeld", "dfbeta", "dfbetas",
    "scaledsch"),
  collapse = FALSE,
  weighted = (type %in% c("dfbeta", "dfbetas"))
)
```

Arguments

`object` The output from the `phregr` call.

`type` The type of residuals desired, with options including `"martingale"`, `"deviance"`, `"score"`, `"schoenfeld"`, `"dfbeta"`, `"dfbetas"`, and `"scaledsch"`.

`collapse` Whether to collapse the residuals by `id`. This is not applicable for Schoenfeld type residuals.

`weighted` Whether to compute weighted residuals.

Details

For score and Schoenfeld type residuals, the proportional hazards model must include at least one covariate. The algorithms for `deviance`, `dfbeta`, `dfbetas`, and `scaledsch` residuals follow the `residuals.coxph` function in the `survival` package.

Value

For martingale and deviance residuals, the result is a vector with one element corresponding to each subject (without `collapse`). For score residuals, the result is a matrix where each row represents a subject and each column corresponds to a variable. The row order aligns with the input data used in the original fit. For Schoenfeld residuals, the result is a matrix with one row for each event and one column per variable. These rows are sorted by time within strata, with the attributes `stratum` and `time` included.

Score residuals represent each individual's contribution to the score vector. Two commonly used transformations of this are `dfbeta`, which represents the approximate change in the coefficient vector if the observation is excluded, and `dfbetas`, which gives the approximate change in the coefficients scaled by the standard error of the coefficients.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Terry M. Therneau, Patricia M. Grambsch, and Thomas M. Fleming. Martingale based residuals for survival models. Biometrika 1990; 77:147-160.

Patricia M. Grambsch and Terry M. Therneau. Proportional hazards tests and diagnostics based on weighted residuals. Biometrika 1994; 81:515-26.

Examples

```
library(dplyr)

# Example 1 with right-censored data
fit1 <- phregr(data = rawdata %>% filter(iterationNumber == 1) %>%
                 mutate(treat = 1*(treatmentGroup == 1)),
               stratum = "stratum",
               time = "timeUnderObservation", event = "event",
               covariates = "treat")
```

```
ressco <- residuals_phregr(fit1, type = "score")
head(ressco)

# Example 2 with counting process data
fit2 <- phregr(data = heart %>% mutate(rx = as.numeric(transplant) - 1),
               time = "start", time2 = "stop", event = "event",
               covariates = c("rx", "age"), id = "id", robust = TRUE)

resssch <- residuals_phregr(fit2, type = "scaledsch")
head(resssch)
```

rmdiff *Estimate of Restricted Mean Survival Time Difference*

Description

Obtains the estimate of restricted mean survival time difference between two treatment groups.

Usage

```
rmdiff(
  data,
  rep = "",
  stratum = "",
  treat = "treat",
  time = "time",
  event = "event",
  milestone = NA_real_,
  rmstDiffH0 = 0,
  conflev = 0.95,
  biascorrection = 0L
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `treat`: The treatment.
- `time`: The possibly right-censored survival time.
- `event`: The event indicator.

`rep` The name of the replication variable in the input data.

`stratum` The name of the stratum variable in the input data.

`treat` The name of the treatment variable in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`milestone` The milestone time at which to calculate the restricted mean survival time.

`rmstDiffH0` The difference in restricted mean survival times under the null hypothesis. Defaults to 0 for superiority test.

`conflev` The level of the two-sided confidence interval for the difference in restricted mean survival times. Defaults to 0.95.

`biascorrection` Whether to apply bias correction for the variance estimate of individual restricted mean survival times. Defaults to no bias correction.

Value

A data frame with the following variables:

- `rep`: The replication number.
- `milestone`: The milestone time relative to randomization.
- `rmstDiffH0`: The difference in restricted mean survival times under the null hypothesis.
- `rmst1`: The estimated restricted mean survival time for the treatment group.
- `rmst2`: The estimated restricted mean survival time for the control group.
- `rmstDiff`: The estimated difference in restricted mean survival times.
- `vrmst1`: The variance for rmst1.
- `vrmst2`: The variance for rmst2.
- `vrmstDiff`: The variance for rmstDiff.
- `rmstDiffZ`: The Z-statistic value.
- `rmstDiffPValue`: The one-sided p-value.
- `lower`: The lower bound of confidence interval.
- `upper`: The upper bound of confidence interval.
- `conflev`: The level of confidence interval.
- `biascorrection`: Whether to apply bias correction for the variance estimate of individual restricted mean survival times.

Author(s)

Kaifeng Lu, `<kaifenglu@gmail.com>`

Examples

```
df <- rmdiff(data = rawdata, rep = "iterationNumber",
             stratum = "stratum", treat = "treatmentGroup",
             time = "timeUnderObservation", event = "event",
             milestone = 12)
head(df)
```

rmest *Estimate of Restricted Mean Survival Time*

Description

Obtains the estimate of restricted means survival time for each stratum.

Usage

```
rmest(
  data,
  rep = "",
  stratum = "",
  time = "time",
  event = "event",
  milestone = NA_real_,
  conflev = 0.95,
  biascorrection = 0L
)
```

Arguments

`data` The input data frame that contains the following variables:

- `rep`: The replication for by-group processing.
- `stratum`: The stratum.
- `time`: The possibly right-censored survival time.
- `event`: The event indicator.

`rep` The name of the replication variable in the input data.

`stratum` The name of the stratum variable in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`milestone` The milestone time at which to calculate the restricted mean survival time.

`conflev` The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.

`biascorrection` Whether to apply bias correction for the variance estimate. Defaults to no bias correction.

Value

A data frame with the following variables:

- `rep`: The replication.
- `stratum`: The stratum variable.
- `size`: The number of subjects in the stratum.

- `milestone`: The milestone time relative to randomization.
- `rmst`: The estimate of restricted mean survival time.
- `stderr`: The standard error of the estimated rmst.
- `lower`: The lower bound of confidence interval if requested.
- `upper`: The upper bound of confidence interval if requested.
- `conflev`: The level of confidence interval if requested.
- `biascorrection`: Whether to apply bias correction for the variance estimate.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
rmest(data = aml, stratum = "x",
      time = "time", event = "status", milestone = 24)
```

rpsftm *Rank Preserving Structural Failure Time Model (RPSFTM) for Treatment Switching*

Description

Obtains the causal parameter estimate from the log-rank test, the Cox proportional hazards model, or a parametric survival regression model, and obtains the hazard ratio estimate from the Cox model to adjust for treatment switching.

Usage

```
rpsftm(
  data,
  id = "id",
  stratum = "",
  time = "time",
  event = "event",
  treat = "treat",
  rx = "rx",
  censor_time = "censor_time",
  base_cov = "",
  psi_test = "logrank",
  aft_dist = "weibull",
  strata_main_effect_only = TRUE,
  low_psi = -2,
  hi_psi = 2,
  n_eval_z = 101,
  treat_modifier = 1,
  recensor = TRUE,
  admin_recensor_only = TRUE,
  autoswitch = TRUE,
  gridsearch = FALSE,
  root_finding = "brent",
  alpha = 0.05,
  ties = "efron",
  tol = 1e-06,
  boot = FALSE,
  n_boot = 1000,
  seed = NA
)
```

Arguments

`data` The input data frame that contains the following variables:

- `id`: The subject id.
- `stratum`: The stratum.
- `time`: The survival time for right censored data.
- `event`: The event indicator, 1=event, 0=no event.
- `treat`: The randomized treatment indicator, 1=treatment, 0=control.
- `rx`: The proportion of time on active treatment.
- `censor_time`: The administrative censoring time. It should be provided for all subjects including those who had events.
- `base_cov`: The baseline covariates (excluding treat).

`id` The name of the id variable in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`time` The name of the time variable in the input data.

`event` The name of the event variable in the input data.

`treat` The name of the treatment variable in the input data.

`rx` The name of the rx variable in the input data.

`censor_time` The name of the censor_time variable in the input data.

`base_cov` The names of baseline covariates (excluding treat) in the input data for the outcome Cox model. These covariates will also be used in the Cox model for estimating `psi` when `psi_test = "phreg"` and in the AFT model for estimating `psi` when `psi_test = "lifereg"`.

`psi_test` The survival function to calculate the Z-statistic, e.g., "logrank" (default), "phreg", or "lifereg".

`aft_dist` The assumed distribution for time to event for the accelerated failure time (AFT) model when `psi_test = "lifereg"`. Options include "exponential", "weibull" (default), "loglogistic", and "lognormal".

`strata_main_effect_only`
Whether to only include the strata main effects in the AFT model. Defaults to `TRUE`, otherwise all possible strata combinations will be considered in the AFT model.

`low_psi` The lower limit of the causal parameter.

`hi_psi` The upper limit of the causal parameter.

`n_eval_z` The number of points between `low_psi` and `hi_psi` (inclusive) at which to evaluate the Z-statistics.

`treat_modifier` The optional sensitivity parameter for the constant treatment effect assumption.

`recensor` Whether to apply recensoring to counterfactual survival times. Defaults to `TRUE`.

`admin_recensor_only`
Whether to apply recensoring to administrative censoring times only. Defaults to `TRUE`. If `FALSE`, recensoring will be applied to the actual censoring times for dropouts.

`autoswitch` Whether to exclude recensoring for treatment arms with no switching. Defaults to `TRUE`.

`gridsearch` Whether to use grid search to estimate the causal parameter `psi`. Defaults to `FALSE`, in which case, a root finding algorithm will be used.

`root_finding` Character string specifying the univariate root-finding algorithm to use. Options are `"brent"` (default) for Brent's method, or `"bisection"` for the bisection method.

`alpha` The significance level to calculate confidence intervals.

`ties` The method for handling ties in the Cox model, either "breslow" or "efron" (default).

`tol` The desired accuracy (convergence tolerance) for `psi` for the root finding algorithm.

`boot` Whether to use bootstrap to obtain the confidence interval for hazard ratio. Defaults to `FALSE`, in which case, the confidence interval will be constructed to match the log-rank test p-value.

`n_boot` The number of bootstrap samples.

`seed` The seed to reproduce the bootstrap results. The default is missing, in which case, the seed from the environment will be used.

Details

We use the following steps to obtain the hazard ratio estimate and confidence interval had there been no treatment switching:

- Use RPSFTM to estimate the causal parameter ψ based on the log-rank test (default), the Cox proportional hazards model, or a parametric survival regression model for counterfactual *untreated* survival times:

$$U_{i,\psi} = T_{C_i} + e^{\psi} T_{E_i}$$

- Fit the Cox proportional hazards model to the counterfactual *unswitched* survival times to obtain the hazard ratio estimate.

- Use either the log-rank test p-value for the intention-to-treat (ITT) analysis or bootstrap to construct the confidence interval for hazard ratio. If bootstrapping is used, the confidence interval and corresponding p-value are calculated based on a t-distribution with `n_boot - 1` degrees of freedom.

Value

A list with the following components:

- `psi`: The estimated causal parameter.
- `psi_CI`: The confidence interval for `psi`.
- `psi_CI_type`: The type of confidence interval for `psi`, i.e., "grid search", "root finding", or "bootstrap".
- `logrank_pvalue`: The two-sided p-value of the log-rank test for the ITT analysis.
- `cox_pvalue`: The two-sided p-value for treatment effect based on the Cox model applied to counterfactual unswitched survival times. If `boot` is `TRUE`, this value represents the bootstrap p-value.
- `hr`: The estimated hazard ratio from the Cox model.
- `hr_CI`: The confidence interval for hazard ratio.
- `hr_CI_type`: The type of confidence interval for hazard ratio, either "log-rank p-value" or "bootstrap".
- `eval_z`: A data frame containing the Z-statistics for treatment effect evaluated at a sequence of `psi` values. Used to plot and check if the range of `psi` values to search for the solution and limits of confidence interval of `psi` need be modified.
- `Sstar`: A data frame containing the counterfactual untreated survival times and event indicators for each treatment group. The variables include `id`, `stratum`, `"t_star"`, `"d_star"`, `"treated"`, `base_cov`, and `treat`.
- `kmstar`: A data frame containing the Kaplan-Meier estimates based on the counterfactual untreated survival times by treatment arm.
- `data_outcome`: The input data for the outcome Cox model of counterfactual unswitched survival times. The variables include `id`, `stratum`, `"t_star"`, `"d_star"`, `"treated"`, `base_cov`, and `treat`.
- `fit_outcome`: The fitted outcome Cox model.
- `fail`: Whether a model fails to converge.
- `psimissing`: Whether the psi parameter cannot be estimated.
- `settings`: A list with the following components:
  - `psi_test`: The survival function to calculate the Z-statistic.
  - `aft_dist`: The distribution for time to event for the AFT model.
  - `strata_main_effect_only`: Whether to only include the strata main effects in the AFT model.
  - `low_psi`: The lower limit of the causal parameter.
  - `hi_psi`: The upper limit of the causal parameter.
  - `n_eval_z`: The number of points between `low_psi` and `hi_psi` (inclusive) at which to evaluate the Z-statistics.

  - treat_modifier: The sensitivity parameter for the constant treatment effect assumption.
  - recensor: Whether to apply recensoring to counterfactual survival times.
  - admin_recensor_only: Whether to apply recensoring to administrative censoring times only.
  - autoswitch: Whether to exclude recensoring for treatment arms with no switching.
  - gridsearch: Whether to use grid search to estimate the causal parameter psi.
  - root_finding: The univariate root-finding algorithm to use.
  - alpha: The significance level to calculate confidence intervals.
  - ties: The method for handling ties in the Cox model.
  - tol: The desired accuracy (convergence tolerance) for psi.
  - boot: Whether to use bootstrap to obtain the confidence interval for hazard ratio.
  - n_boot: The number of bootstrap samples.
  - seed: The seed to reproduce the bootstrap results.
- fail_boots: The indicators for failed bootstrap samples if boot is TRUE.
- fail_boots_data: The data for failed bootstrap samples if boot is TRUE.
- hr_boots: The bootstrap hazard ratio estimates if boot is TRUE.
- psi_boots: The bootstrap psi estimates if boot is TRUE.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

James M. Robins and Anastasios A. Tsiatis. Correcting for non-compliance in randomized trials using rank preserving structural failure time models. Communications in Statistics. 1991;20(8):2609-2631.

Ian R. White, Adbel G. Babiker, Sarah Walker, and Janet H. Darbyshire. Randomization-based methods for correcting for treatment changes: Examples from the CONCORDE trial. Statistics in Medicine. 1999;18(19):2617-2634.

Examples

```
library(dplyr)

# Example 1: one-way treatment switching (control to active)

data <- immdef %>% mutate(rx = 1-xoyrs/progyrs)

fit1 <- rpsftm(
  data, id = "id", time = "progyrs", event = "prog", treat = "imm",
  rx = "rx", censor_time = "censyrs", boot = FALSE)

c(fit1$hr, fit1$hr_CI)

# Example 2: two-way treatment switching (illustration only)
# the eventual survival time
shilong1 <- shilong %>%
  arrange(bras.f, id, tstop) %>%
  group_by(bras.f, id) %>%
  slice(n()) %>%
  select(-c("ps", "ttc", "tran"))

shilong2 <- shilong1 %>%
  mutate(rx = ifelse(co, ifelse(bras.f == "MTA", dco/ady,
                                1 - dco/ady),
                     ifelse(bras.f == "MTA", 1, 0)))

fit2 <- rpsftm(
  shilong2, id = "id", time = "tstop", event = "event",
  treat = "bras.f", rx = "rx", censor_time = "dcut",
  base_cov = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
               "pathway.f"),
  low_psi = -3, hi_psi = 3, boot = FALSE)

c(fit2$hr, fit2$hr_CI)
```

sexagg *Urinary tract infection data from the logistf package*

Description

This data set deals with urinary tract infection in sexually active college women, along with covariate information on age an contraceptive use. The variables are all binary and coded in 1 (condition is present) and 0 (condition is absent).

Usage

```
sexagg
```

Format

An object of class `data.frame` with 36 rows and 9 columns.

Details

`case` urinary tract infection, the study outcome variable

`age` >= 24 years

`dia` use of diaphragm

`oc` use of oral contraceptive

`vic` use of condom

`vicl` use of lubricated condom

`vis` use of spermicide

`shilong` *The randomized clinical trial SHIVA data in long format from the ipcwswitch package*

Description

The original SHIdat data set contains an anonymized excerpt of data from the SHIVA01 trial. This was the first randomized clinical trial that aimed at comparing molecularly targeted therapy based on tumor profiling (MTA) versus conventional therapy (CT) for advanced cancer. Patients were randomly assigned to receive the active or control treatment and may switch to the other arm or subsequent anti-cancer therapy upon disease progression. The restructured data is in the long format.

`id` The patient's identifier

`tstart` The start of the time interval

`tstop` The end of the time interval

`event` Whether the patient died at the end of the interval

`agerand` The patient's age (in years) at randomization

`sex.f` The patients' gender, either Male or Female

`tt_Lnum` The number of previous lines of treatment

`rmh_alea.c` The Royal Marsden Hospital score segregated into two categories

`pathway.f` The molecular pathway altered (the hormone receptors pathway, the PI3K/ AKT/mTOR pathway, and the RAF/MEK pathway)

`bras.f` The patient's randomized arm, either MTA or CT

`ps` The ECOG performance status

`ttc` The presence of concomitant treatments

`tran` The use of platelet transfusions

`dpd` The relative day of a potential progression

`dco` The relative day of treatment switching

`ady` The relative day of the latest news

`dcut` The relative day of administrative cutoff

`pd` Whether the patient had disease progression

`co` Whether the patient switched treatment

Usage

```
shilong
```

Format

An object of class `data.frame` with 602 rows and 19 columns.

six *The repeated measures data from the "Six Cities" study of the health effects of air pollution (Ware et al. 1984).*

Description

The data analyzed are the 16 selected cases in Lipsitz et al. (1994). The binary response is the wheezing status of 16 children at ages 9, 10, 11, and 12 years. A value of 1 of wheezing status indicates the occurrence of wheezing. The explanatory variables city of residence, age, and maternal smoking status at the particular age.

Usage

```
six
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 64 rows and 6 columns.

Details

`case` case id

`city` city of residence

`age` age of the child

`smoke` maternal smoking status

`wheeze` wheezing status

splineDesigncpp *B-Spline Design Matrix*

Description

Computes the design matrix for B-splines based on the specified `knots` and evaluated at the values in `x`.

Usage

```
splineDesigncpp(
  knots = NA_real_,
  x = NA_real_,
  ord = 4L,
  derivs = as.integer(c(0))
)
```

Arguments

`knots`	A numeric vector specifying the positions of the knots, including both boundary and internal knots.
`x`	A numeric vector of values where the B-spline functions or their derivatives will be evaluated. The values of `x` must lie within the range of the "inner" knots, i.e., between `knots[ord]` and `knots[length(knots) - (ord - 1)]`.
`ord`	A positive integer indicating the order of the B-spline. This corresponds to the number of coefficients in each piecewise polynomial segment, where `ord = degree + 1`.
`derivs`	An integer vector specifying the order of derivatives to be evaluated at the corresponding `x` values. Each value must be between `0` and `ord - 1`, and the vector is conceptually recycled to match the length of `x`. The default is `0`, meaning the B-spline functions themselves are evaluated.

Value

A matrix with dimensions `c(length(x), length(knots) - ord)`. Each row corresponds to a value in `x` and contains the coefficients of the B-splines, or the specified derivatives, as defined by the `knots` and evaluated at that particular value of `x`. The total number of B-splines is `length(knots) - ord`, with each B-spline defined by a set of `ord` consecutive knots.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

Examples

```
splineDesigncpp(knots = 1:10, x = 4:7)
splineDesigncpp(knots = 1:10, x = 4:7, derivs = 1)
```

survfit_phregr *Survival Curve for Proportional Hazards Regression Models*

Description

Obtains the predicted survivor function for a proportional hazards regression model.

Usage

```
survfit_phregr(
  object,
  newdata,
  sefit = TRUE,
  conftype = "log-log",
  conflev = 0.95
)
```

Arguments

`object` The output from the `phregr` call.

`newdata` A data frame with the same variable names as those that appear in the `phregr` call. For right-censored data, one curve is produced per row to represent a cohort whose covariates correspond to the values in `newdata`. For counting-process data, one curve is produced per `id` in `newdata` to present the survival curve along the path of time-dependent covariates at the observed event times in the data used to fit `phregr`.

`sefit` Whether to compute the standard error of the survival estimates.

`conftype` The type of the confidence interval. One of `"none"`, `"plain"`, `"log"`, `"log-log"` (the default), or `"arcsin"`. The `arcsin` option bases the intervals on `asin(sqrt(surv))`.

`conflev` The level of the two-sided confidence interval for the survival probabilities. Defaults to 0.95.

Details

If `newdata` is not provided and there is no covariate, survival curves based on the `basehaz` data frame will be produced.

Value

A data frame with the following variables:

- `id`: The id of the subject for counting-process data with time-dependent covariates.
- `time`: The observed times in the data used to fit `phregr`.
- `nrisk`: The number of patients at risk at the time point in the data used to fit `phregr`.
- `nevent`: The number of patients having event at the time point in the data used to fit `phregr`.
- `cumhaz`: The cumulative hazard at the time point.
- `surv`: The estimated survival probability at the time point.
- `sesurv`: The standard error of the estimated survival probability.
- `lower`: The lower confidence limit for survival probability.
- `upper`: The upper confidence limit for survival probability.
- `conflev`: The level of the two-sided confidence interval.
- `conftype`: The type of the confidence interval.
- `covariates`: The values of covariates based on `newdata`.
- `stratum`: The stratum of the subject.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

References

Terry M. Therneau and Patricia M. Grambsch. Modeling Survival Data: Extending the Cox Model. Springer-Verlag, 2000.

Examples

```
library(dplyr)

# Example 1 with right-censored data
fit1 <- phregr(data = rawdata %>% filter(iterationNumber == 1) %>%
                 mutate(treat = 1*(treatmentGroup == 1)),
               stratum = "stratum",
               time = "timeUnderObservation", event = "event",
               covariates = "treat")

surv1 <- survfit_phregr(fit1,
                        newdata = data.frame(
                          stratum = as.integer(c(1,1,2,2)),
                          treat = c(1,0,1,0)))
head(surv1)

# Example 2 with counting process data and robust variance estimate
fit2 <- phregr(data = heart %>% mutate(rx = as.numeric(transplant) - 1),
               time = "start", time2 = "stop", event = "event",
               covariates = c("rx", "age"), id = "id", robust = TRUE)

surv2 <- survfit_phregr(fit2,
                        newdata = data.frame(
                          id = c(4,4,11,11),
                          age = c(-7.737,-7.737,-0.019,-0.019),
                          start = c(0,36,0,26),
                          stop = c(36,39,26,153),
                          rx = c(0,1,0,1)))
head(surv2)
```

survQuantile *Brookmeyer-Crowley Confidence Interval for Quantiles of Right-Censored Time-to-Event Data*

Description

Obtains the Brookmeyer-Crowley confidence interval for quantiles of right-censored time-to-event data.

Usage

```
survQuantile(
  time = NA_real_,
  event = NA_real_,
  cilevel = 0.95,
  transform = "loglog",
  probs = NA_real_
)
```

Arguments

`time` The vector of possibly right-censored survival times.

`event` The vector of event indicators.

`cilevel` The confidence interval level. Defaults to 0.95.

`transform` The transformation of the survival function to use to construct the confidence interval. Options include "linear" (alternatively "plain"), "log", "loglog" (alternatively "log-log" or "cloglog"), "asinsqrt" (alternatively "asin" or "arcsin"), and "logit". Defaults to "loglog".

`probs` The vector of probabilities to calculate the quantiles. Defaults to c(0.25, 0.5, 0.75).

Value

A data frame containing the estimated quantile and confidence interval corresponding to each specified probability. It includes the following variables:

- `prob`: The probability to calculate the quantile.
- `quantile`: The estimated quantile.
- `lower`: The lower limit of the confidence interval.
- `upper`: The upper limit of the confidence interval.
- `cilevel`: The confidence interval level.
- `transform`: The transformation of the survival function to use to construct the confidence interval.

Author(s)

Kaifeng Lu, <`kaifenglu@gmail.com`>

Examples

```
survQuantile(
  time = c(33.7, 3.9, 10.5, 5.4, 19.5, 23.8, 7.9, 16.9, 16.6,
           33.7, 17.1, 7.9, 10.5, 38),
  event = c(0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1),
  probs = c(0.25, 0.5, 0.75))
```

`tobin` *Tobin's tobit data from the survival package*

Description

Data from Tobin's original paper.

`durable` Durable goods purchase

`age` Age in years

`quant` Liquidity ratio (x 1000)

Usage

```
tobin
```

Format

An object of class `data.frame` with 20 rows and 3 columns.

`tsegest` *The Two-Stage Estimation (TSE) Method Using g-estimation for Treatment Switching*

Description

Obtains the causal parameter estimate using g-estimation based on the logistic regression switching model and the hazard ratio estimate of the Cox model to adjust for treatment switching.

Usage

```
tsegest(
  data,
  id = "id",
  stratum = "",
  tstart = "tstart",
  tstop = "tstop",
  event = "event",
  treat = "treat",
  censor_time = "censor_time",
  pd = "pd",
  pd_time = "pd_time",
  swtrt = "swtrt",
  swtrt_time = "swtrt_time",
  base_cov = "",
  conf_cov = "",
  low_psi = -2,
  hi_psi = 2,
  n_eval_z = 101,
  strata_main_effect_only = TRUE,
  firth = FALSE,
  flic = FALSE,
  ns_df = 3,
  recensor = TRUE,
  admin_recensor_only = TRUE,
  swtrt_control_only = TRUE,
  gridsearch = FALSE,
  root_finding = "brent",
  alpha = 0.05,
  ties = "efron",
  tol = 1e-06,
  offset = 1,
  boot = TRUE,
  n_boot = 1000,
  seed = NA
)
```

Arguments

`data` The input data frame that contains the following variables:

- `id`: The id to identify observations belonging to the same subject for counting process data with time-dependent covariates.
- `stratum`: The stratum.
- `tstart`: The starting time of the time interval for counting-process data with time-dependent covariates.
- `tstop`: The stopping time of the time interval for counting-process data with time-dependent covariates.
- `event`: The event indicator, 1=event, 0=no event.
- `treat`: The randomized treatment indicator, 1=treatment, 0=control.
- `censor_time`: The administrative censoring time. It should be provided for all subjects including those who had events.
- `pd`: The disease progression indicator, 1=PD, 0=no PD.
- `pd_time`: The time from randomization to PD.
- `swtrt`: The treatment switch indicator, 1=switch, 0=no switch.
- `swtrt_time`: The time from randomization to treatment switch.
- `base_cov`: The baseline covariates (excluding treat).
- `conf_cov`: The confounding variables for predicting treatment switching (excluding treat).

`id` The name of the id variable in the input data.

`stratum` The name(s) of the stratum variable(s) in the input data.

`tstart` The name of the tstart variable in the input data.

`tstop` The name of the tstop variable in the input data.

`event` The name of the event variable in the input data.

`treat` The name of the treatment variable in the input data.

`censor_time` The name of the censor_time variable in the input data.

`pd` The name of the pd variable in the input data.

`pd_time` The name of the pd_time variable in the input data.

`swtrt` The name of the swtrt variable in the input data.

`swtrt_time` The name of the swtrt_time variable in the input data.

`base_cov` The names of baseline covariates (excluding treat) in the input data for the Cox model.

`conf_cov` The names of confounding variables (excluding treat) in the input data for the logistic regression switching model.

`low_psi` The lower limit of the causal parameter.

`hi_psi` The upper limit of the causal parameter.

`n_eval_z` The number of points between `low_psi` and `hi_psi` (inclusive) at which to evaluate the Wald statistics for the coefficient of the counterfactual in the logistic regression switching model.

`strata_main_effect_only`
Whether to only include the strata main effects in the logistic regression switching model. Defaults to `TRUE`, otherwise all possible strata combinations will be considered in the switching model.

`firth` Whether the Firth's bias reducing penalized likelihood should be used.

`flic` Whether to apply intercept correction to obtain more accurate predicted probabilities.

`ns_df` Degrees of freedom for the natural cubic spline for visit-specific intercepts of the pooled logistic regression model. Defaults to 3 for two internal knots at the 33 and 67 percentiles of the treatment switching times.

`recensor` Whether to apply recensoring to counterfactual survival times. Defaults to `TRUE`.

`admin_recensor_only`
Whether to apply recensoring to administrative censoring times only. Defaults to `TRUE`. If `FALSE`, recensoring will be applied to the actual censoring times for dropouts.

`swtrt_control_only`
Whether treatment switching occurred only in the control group. The default is `TRUE`.

`gridsearch` Whether to use grid search to estimate the causal parameter `psi`. Defaults to `FALSE`, in which case, a root finding algorithm will be used.

`root_finding` Character string specifying the univariate root-finding algorithm to use. Options are `"brent"` (default) for Brent's method, or `"bisection"` for the bisection method.

`alpha` The significance level to calculate confidence intervals. The default value is 0.05.

`ties` The method for handling ties in the Cox model, either "breslow" or "efron" (default).

`tol` The desired accuracy (convergence tolerance) for `psi` for the root finding algorithm.

`offset` The offset to calculate the time to event, PD, and treatment switch. We can set `offset` equal to 1 (default), 1/30.4375, or 1/365.25 if the time unit is day, month, or year.

`boot` Whether to use bootstrap to obtain the confidence interval for hazard ratio. Defaults to `TRUE`.

`n_boot` The number of bootstrap samples.

`seed` The seed to reproduce the bootstrap results. The default is missing, in which case, the seed from the environment will be used.

Details

We use the following steps to obtain the hazard ratio estimate and confidence interval had there been no treatment switching:

- Use a pooled logistic regression switching model to estimate the causal parameter ψ based on the patients in the control group who had disease progression:

  $$\text{logit}(p(E_{ik})) = \alpha U_{i,\psi} + \sum_j \beta_j x_{ijk}$$

  where E_{ik} is the observed switch indicator for individual i at observation k,

  $$U_{i,\psi} = T_{C_i} + e^{\psi} T_{E_i}$$

  is the counterfactual survival time for individual i given a specific value for ψ, and x_{ijk} is the confounder j for individual i at observation k. When applied from a secondary baseline, $U_{i,\psi}$ refers to post-secondary baseline counterfactual survival, where T_{C_i} corresponds to the time spent after the secondary baseline on control treatment, and T_{E_i} corresponds to the time spent after the secondary baseline on the experimental treatment.
- Search for ψ such that the Z-statistic for α is close to zero. This will be the estimate of the causal parameter. The confidence interval for ψ can be obtained as the value of ψ such that the corresponding two-sided p-value for testing $H_0 : \alpha = 0$ in the switching model is equal to the nominal significance level.
- Derive the counterfactual survival times for control patients had there been no treatment switching.
- Fit the Cox proportional hazards model to the observed survival times for the experimental group and the counterfactual survival times for the control group to obtain the hazard ratio estimate.
- If bootstrapping is used, the confidence interval and corresponding p-value for hazard ratio are calculated based on a t-distribution with `n_boot - 1` degrees of freedom.

Value

A list with the following components:

- `psi`: The estimated causal parameter for the control group.
- `psi_CI`: The confidence interval for `psi`.
- `psi_CI_type`: The type of confidence interval for `psi`, i.e., "grid search", "root finding", or "bootstrap".
- `logrank_pvalue`: The two-sided p-value of the log-rank test for the ITT analysis.
- `cox_pvalue`: The two-sided p-value for treatment effect based on the Cox model applied to counterfactual unswitched survival times. If `boot` is `TRUE`, this value represents the bootstrap p-value.
- `hr`: The estimated hazard ratio from the Cox model.
- `hr_CI`: The confidence interval for hazard ratio.
- `hr_CI_type`: The type of confidence interval for hazard ratio, either "Cox model" or "bootstrap".

- analysis_switch: A list of data and analysis results related to treatment switching.
  - data_switch: The list of input data for the time from secondary baseline to switch by treatment group. The variables include id, stratum, "swtrt", and "swtrt_time". If swtrt == 0, then swtrt_time is censored at the time from secondary baseline to either death or censoring.
  - km_switch: The list of Kaplan-Meier plot data for the time from secondary baseline to switch by treatment group.
  - eval_z: The list of data by treatment group containing the Wald statistics for the coefficient of the counterfactual in the logistic regression switching model, evaluated at a sequence of psi values. Used to plot and check if the range of psi values to search for the solution and limits of confidence interval of psi need be modified.
  - data_nullcox: The list of input data for counterfactual survival times for the null Cox model by treatment group. The variables include id, stratum, "t_star" and "d_star".
  - fit_nullcox: The list of fitted null Cox models for counterfactual survival times by treatment group, which contains the martingale residuals.
  - data_logis: The list of input data for pooled logistic regression models for treatment switching using g-estimation. The variables include id, stratum, "tstart", "tstop", "cross", "counterfactual", conf_cov, pd_time, swtrt, and swtrt_time.
  - fit_logis: The list of fitted pooled logistic regression models for treatment switching using g-estimation.
- data_outcome: The input data for the outcome Cox model of counterfactual unswitched survival times. The variables include id, stratum, "t_star", "d_star", "treated", base_cov and treat.
- fit_outcome: The fitted outcome Cox model.
- fail: Whether a model fails to converge.
- psimissing: Whether the psi parameter cannot be estimated.
- settings: A list with the following components:
  - low_psi: The lower limit of the causal parameter.
  - hi_psi: The upper limit of the causal parameter.
  - n_eval_z: The number of points between low_psi and hi_psi (inclusive) at which to evaluate the Wald statistics for the coefficient for the counterfactual in the logistic regression switching model.
  - strata_main_effect_only: Whether to only include the strata main effects in the logistic regression switching model.
  - firth: Whether the Firth's penalized likelihood is used.
  - flic: Whether to apply intercept correction.
  - ns_df: Degrees of freedom for the natural cubic spline.
  - recensor: Whether to apply recensoring to counterfactual survival times.
  - admin_recensor_only: Whether to apply recensoring to administrative censoring times only.
  - swtrt_control_only: Whether treatment switching occurred only in the control group.
  - gridsearch: Whether to use grid search to estimate the causal parameter psi.
  - root_finding: The univariate root-finding algorithm to use.
  - alpha: The significance level to calculate confidence intervals.

  - ties: The method for handling ties in the Cox model.
  - tol: The desired accuracy (convergence tolerance) for psi.
  - offset: The offset to calculate the time to event, PD, and treatment switch.
  - boot: Whether to use bootstrap to obtain the confidence interval for hazard ratio.
  - n_boot: The number of bootstrap samples.
  - seed: The seed to reproduce the bootstrap results.
- psi_trt: The estimated causal parameter for the experimental group if swtrt_control_only is FALSE.
- psi_trt_CI: The confidence interval for psi_trt if swtrt_control_only is FALSE.
- fail_boots: The indicators for failed bootstrap samples if boot is TRUE.
- fail_boots_data: The data for failed bootstrap samples if boot is TRUE.
- hr_boots: The bootstrap hazard ratio estimates if boot is TRUE.
- psi_boots: The bootstrap psi estimates if boot is TRUE.
- psi_trt_boots: The bootstrap psi_trt estimates if boot is TRUE and swtrt_control_only is FALSE.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

NR Latimer, IR White, K Tilling, and U Siebert. Improved two-stage estimation to adjust for treatment switching in randomised trials: g-estimation to address time-dependent confounding. Statistical Methods in Medical Research. 2020;29(10):2900-2918.

Examples

```
library(dplyr)

sim1 <- tsegestsim(
  n = 500, allocation1 = 2, allocation2 = 1, pbprog = 0.5,
  trtlghr = -0.5, bprogsl = 0.3, shape1 = 1.8,
  scale1 = 360, shape2 = 1.7, scale2 = 688,
  pmix = 0.5, admin = 5000, pcatnotrtbprog = 0.5,
  pcattrtbprog = 0.25, pcatnotrt = 0.2, pcattrt = 0.1,
  catmult = 0.5, tdxo = 1, ppoor = 0.1, pgood = 0.04,
  ppoormet = 0.4, pgoodmet = 0.2, xomult = 1.4188308,
  milestone = 546, outputRawDataset = 1, seed = 2000)

data1 <- sim1$paneldata %>%
  mutate(visit7on = ifelse(progressed, tstop > timePFSobs + 105, 0))

fit1 <- tsegest(
  data = data1, id = "id",
  tstart = "tstart", tstop = "tstop", event = "event",
  treat = "trtrand", censor_time = "censor_time",
  pd = "progressed", pd_time = "timePFSobs",
  swtrt = "xo", swtrt_time = "xotime",
  base_cov = "bprog",
  conf_cov = c("bprog*cattdc", "timePFSobs", "visit7on"),
  ns_df = 3, recensor = TRUE, admin_recensor_only = TRUE,
  swtrt_control_only = TRUE, alpha = 0.05, ties = "efron",
  tol = 1.0e-6, offset = 0, boot = FALSE)

c(fit1$hr, fit1$hr_CI)
```

tsegestsim *Simulate Survival Data for Two-Stage Estimation Method Using g-estimation*

Description

Obtains the simulated data for baseline prognosis, disease progression, treatment switching, death, and time-dependent covariates.

Usage

```
tsegestsim(
  n = 500L,
  allocation1 = 2L,
  allocation2 = 1L,
  pbprog = 0.5,
  trtlghr = -0.5,
  bprogsl = 0.3,
  shape1 = 1.8,
  scale1 = 360,
  shape2 = 1.7,
  scale2 = 688,
  pmix = 0.5,
  admin = 5000,
  pcatnotrtbprog = 0.5,
  pcattrtbprog = 0.25,
  pcatnotrt = 0.2,
  pcattrt = 0.1,
  catmult = 0.5,
  tdxo = 1,
  ppoor = 0.1,
  pgood = 0.04,
  ppoormet = 0.4,
  pgoodmet = 0.2,
  xomult = 1.4188308,
  milestone = 546,
  outputRawDataset = 1L,
  seed = NA_integer_
)
```

Arguments

`n` The total sample size for two treatment arms combined.

`allocation1` The number of subjects in the active treatment group in a randomization block.

`allocation2` The number of subjects in the control group in a randomization block.

`pbprog` The probability of having poor prognosis at baseline.

`trtlghr` The treatment effect in terms of log hazard ratio.

`bprogsl` The poor prognosis effect in terms of log hazard ratio.

`shape1` The shape parameter for the Weibull event distribution for the first component.

`scale1` The scale parameter for the Weibull event distribution for the first component.

`shape2` The shape parameter for the Weibull event distribution for the second component.

`scale2` The scale parameter for the Weibull event distribution for the second component.

`pmix` The mixing probability of the first component Weibull distribution.

`admin` The administrative censoring time.

`pcatnotrtbprog` The probability of developing metastatic disease on control treatment with poor baseline prognosis.

`pcattrtbprog` The probability of developing metastatic disease on active treatment with poor baseline prognosis.

`pcatnotrt` The probability of developing metastatic disease on control treatment with good baseline prognosis.

`pcattrt` The probability of developing metastatic disease on active treatment with good baseline prognosis.

`catmult` The impact of metastatic disease on shortening remaining survival time.

`tdxo` Whether treatment crossover depends on time-dependent covariates between disease progression and treatment switching.

`ppoor` The probability of switching for poor baseline prognosis with no metastatic disease.

`pgood` The probability of switching for good baseline prognosis with no metastatic disease.

`ppoormet` The probability of switching for poor baseline prognosis after developing metastatic disease.

`pgoodmet` The probability of switching for good baseline prognosis after developing metastatic disease.

`xomult` The direct effect of crossover on extending remaining survival time.

`milestone` The milestone to calculate restricted mean survival time.

`outputRawDataset`
Whether to output the raw data set.

`seed` The seed to reproduce the simulation results. The seed from the environment will be used if left unspecified.

Value

A list with two data frames.

- `sumdata`: A summary data frame with the following variables:
  - `simtrueconstmean`: The true control group restricted mean survival time (RMST).
  - `simtrueconstlb`: The lower bound for control group RMST.
  - `simtrueconstub`: The upper bound for control group RMST.
  - `simtrueconstse`: The standard error for control group RMST.
  - `simtrueexpstmean`: The true experimental group restricted mean survival time (RMST).
  - `simtrueexpstlb`: The lower bound for experimental group RMST.
  - `simtrueexpstub`: The upper bound for experimental group RMST.
  - `simtrueexpstse`: The standard error for experimental group RMST.
  - `simtrue_coxwbprog_hr`: The treatment hazard ratio from the Cox model adjusting for baseline prognosis.
  - `simtrue_cox_hr`: The treatment hazard ratio from the Cox model without adjusting for baseline prognosis.
  - `simtrue_aftwbprog_af`: The average acceleration factor from the Weibull AFT model adjusting for baseline prognosis.
  - `simtrue_aft_af`: The average acceleration factor from the Weibull AFT model without adjusting for baseline prognosis.
- `paneldata`: A counting process style subject-level data frame with the following variables:
  - `id`: The subject ID.
  - `trtrand`: The randomized treatment arm.
  - `bprog`: Whether the patient had poor baseline prognosis.
  - `tstart`: The left end of time interval.
  - `tstop`: The right end of time interval.
  - `event`: Whether the patient died at the end of the interval.
  - `timeOS`: The observed survival time.
  - `died`: Whether the patient died during the study.
  - `progressed`: Whether the patient had disease progression.
  - `timePFSobs`: The observed time of disease progression at regular scheduled visits.
  - `progtdc`: The time-dependent covariate for progression.
  - `catevent`: Whether the patient developed metastatic disease.
  - `cattime`: When the patient developed metastatic disease.
  - `cattdc`: The time-dependent covariate for cat event.
  - `xo`: Whether the patient switched treatment.
  - `xotime`: When the patient switched treatment.
  - `xotdc`: The time-dependent covariate for treatment switching.
  - `censor_time`: The administrative censoring time.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

NR Latimer, IR White, K Tilling, and U Siebert. Improved two-stage estimation to adjust for treatment switching in randomised trials: g-estimation to address time-dependent confounding. Statistical Methods in Medical Research. 2020;29(10):2900-2918.

Examples

```
sim1 <- tsegestsim(
  n = 500, allocation1 = 2, allocation2 = 1, pbprog = 0.5,
  trtlghr = -0.5, bprogsl = 0.3, shape1 = 1.8,
  scale1 = 360, shape2 = 1.7, scale2 = 688,
  pmix = 0.5, admin = 5000, pcatnotrtbprog = 0.5,
  pcattrtbprog = 0.25, pcatnotrt = 0.2, pcattrt = 0.1,
  catmult = 0.5, tdxo = 1, ppoor = 0.1, pgood = 0.04,
  ppoormet = 0.4, pgoodmet = 0.2, xomult = 1.4188308,
  milestone = 546, outputRawDataset = 1, seed = 2000)
```

tsesimp *The Simple Two-Stage Estimation (TSE) Method for Treatment Switching*

Description

Obtains the causal parameter estimate of the accelerated failure-time (AFT) model for switching after disease progression and the hazard ratio estimate of the outcome Cox model to adjust for treatment switching.

Usage

```
tsesimp(
  data,
  id = "id",
  stratum = "",
  time = "time",
  event = "event",
  treat = "treat",
  censor_time = "censor_time",
  pd = "pd",
  pd_time = "pd_time",
  swtrt = "swtrt",
  swtrt_time = "swtrt_time",
  base_cov = "",
  base2_cov = "",
  aft_dist = "weibull",
  strata_main_effect_only = TRUE,
  recensor = TRUE,
  admin_recensor_only = TRUE,
  swtrt_control_only = TRUE,
  alpha = 0.05,
  ties = "efron",
  offset = 1,
  boot = TRUE,
  n_boot = 1000,
  seed = NA
)
```

Arguments

data The input data frame that contains the following variables:

- id: The subject id.
- stratum: The stratum.
- time: The survival time for right censored data.
- event: The event indicator, 1=event, 0=no event.
- treat: The randomized treatment indicator, 1=treatment, 0=control.
- censor_time: The administrative censoring time. It should be provided for all subjects including those who had events.
- pd: The disease progression indicator, 1=PD, 0=no PD.
- pd_time: The time from randomization to PD.
- swtrt: The treatment switch indicator, 1=switch, 0=no switch.
- swtrt_time: The time from randomization to treatment switch.
- base_cov: The baseline covariates (excluding treat).
- base2_cov: The baseline and secondary baseline covariates (excluding swtrt).

id The name of the id variable in the input data.

stratum The name(s) of the stratum variable(s) in the input data.

time The name of the time variable in the input data.

event The name of the event variable in the input data.

treat The name of the treatment variable in the input data.

censor_time The name of the censor_time variable in the input data.

pd The name of the pd variable in the input data.

pd_time The name of the pd_time variable in the input data.

swtrt The name of the swtrt variable in the input data.

swtrt_time The name of the swtrt_time variable in the input data.

base_cov The names of baseline covariates (excluding treat) in the input data for the outcome Cox model.

base2_cov The names of baseline and secondary baseline covariates (excluding swtrt) in the input data for the AFT model for post-progression survival.

aft_dist The assumed distribution for time to event for the AFT model. Options include "exponential", "weibull" (default), "loglogistic", and "lognormal".

strata_main_effect_only
: Whether to only include the strata main effects in the AFT model. Defaults to TRUE, otherwise all possible strata combinations will be considered in the AFT model.

recensor
: Whether to apply recensoring to counterfactual survival times. Defaults to TRUE.

admin_recensor_only
: Whether to apply recensoring to administrative censoring times only. Defaults to TRUE. If FALSE, recensoring will be applied to the actual censoring times for dropouts.

swtrt_control_only
: Whether treatment switching occurred only in the control group. The default is TRUE.

alpha
: The significance level to calculate confidence intervals. The default value is 0.05.

ties
: The method for handling ties in the Cox model, either "breslow" or "efron" (default).

offset
: The offset to calculate the time to event, PD, and treatment switch. We can set offset equal to 1 (default), 1/30.4375, or 1/365.25 if the time unit is day, month, or year.

boot
: Whether to use bootstrap to obtain the confidence interval for hazard ratio. Defaults to TRUE.

n_boot
: The number of bootstrap samples.

seed
: The seed to reproduce the bootstrap results. The default is missing, in which case, the seed from the environment will be used.

Details

We use the following steps to obtain the hazard ratio estimate and confidence interval had there been no treatment switching:

- Fit an AFT model to post-progression survival data to estimate the causal parameter ψ based on the patients in the control group who had disease progression.
- Derive the counterfactual survival times for control patients had there been no treatment switching.
- Fit the Cox proportional hazards model to the observed survival times for the experimental group and the counterfactual survival times for the control group to obtain the hazard ratio estimate.
- If bootstrapping is used, the confidence interval and corresponding p-value for hazard ratio are calculated based on a t-distribution with n_boot - 1 degrees of freedom.

Value

A list with the following components:

- psi: The estimated causal parameter for the control group.
- psi_CI: The confidence interval for psi.

- psi_CI_type: The type of confidence interval for psi, i.e., "AFT model" or "bootstrap".
- logrank_pvalue: The two-sided p-value of the log-rank test for the ITT analysis.
- cox_pvalue: The two-sided p-value for treatment effect based on the Cox model applied to counterfactual unswitched survival times. If boot is TRUE, this value represents the bootstrap p-value.
- hr: The estimated hazard ratio from the Cox model.
- hr_CI: The confidence interval for hazard ratio.
- hr_CI_type: The type of confidence interval for hazard ratio, either "Cox model" or "bootstrap".
- data_aft: A list of input data for the AFT model by treatment group. The variables include id, stratum, "pps", "event", "swtrt", base2_cov, pd_time, swtrt_time, and time.
- fit_aft: A list of fitted AFT models by treatment group.
- data_outcome: The input data for the outcome Cox model of counterfactual unswitched survival times. The variables include id, stratum, "t_star", "d_star", "treated", base_cov, and treat.
- fit_outcome: The fitted outcome Cox model.
- fail: Whether a model fails to converge.
- psimissing: Whether the psi parameter cannot be estimated.
- settings: A list with the following components:
  - aft_dist: The distribution for time to event for the AFT model.
  - strata_main_effect_only: Whether to only include the strata main effects in the AFT model.
  - recensor: Whether to apply recensoring to counterfactual survival times.
  - admin_recensor_only: Whether to apply recensoring to administrative censoring times only.
  - swtrt_control_only: Whether treatment switching occurred only in the control group.
  - alpha: The significance level to calculate confidence intervals.
  - ties: The method for handling ties in the Cox model.
  - offset: The offset to calculate the time to event, PD, and treatment switch.
  - boot: Whether to use bootstrap to obtain the confidence interval for hazard ratio.
  - n_boot: The number of bootstrap samples.
  - seed: The seed to reproduce the bootstrap results.
- psi_trt: The estimated causal parameter for the experimental group if swtrt_control_only is FALSE.
- psi_trt_CI: The confidence interval for psi_trt if swtrt_control_only is FALSE.
- fail_boots: The indicators for failed bootstrap samples if boot is TRUE.
- fail_boots_data: The data for failed bootstrap samples if boot is TRUE.
- hr_boots: The bootstrap hazard ratio estimates if boot is TRUE.
- psi_boots: The bootstrap psi estimates if boot is TRUE.
- psi_trt_boots: The bootstrap psi_trt estimates if boot is TRUE and swtrt_control_only is FALSE.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Nicholas R Latimer, KR Abrams, PC Lambert, MK Crowther, AJ Wailoo, JP Morden, RL Akehurst, and MJ Campbell. Adjusting for treatment switching in randomised controlled trials - A simulation study and a simplified two-stage method. Statistical Methods in Medical Research. 2017;26(2):724-751.

Examples

```
library(dplyr)

# the eventual survival time
shilong1 <- shilong %>%
  arrange(bras.f, id, tstop) %>%
  group_by(bras.f, id) %>%
  slice(n()) %>%
  select(-c("ps", "ttc", "tran"))

# the last value of time-dependent covariates before pd
shilong2 <- shilong %>%
  filter(pd == 0 | tstart <= dpd) %>%
  arrange(bras.f, id, tstop) %>%
  group_by(bras.f, id) %>%
  slice(n()) %>%
  select(bras.f, id, ps, ttc, tran)

# combine baseline and time-dependent covariates
shilong3 <- shilong1 %>%
  left_join(shilong2, by = c("bras.f", "id"))

# apply the two-stage method
fit1 <- tsesimp(
  data = shilong3, id = "id", time = "tstop", event = "event",
  treat = "bras.f", censor_time = "dcut", pd = "pd",
  pd_time = "dpd", swtrt = "co", swtrt_time = "dco",
  base_cov = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
               "pathway.f"),
  base2_cov = c("agerand", "sex.f", "tt_Lnum", "rmh_alea.c",
               "pathway.f", "ps", "ttc", "tran"),
  aft_dist = "weibull", alpha = 0.05,
  recensor = TRUE, swtrt_control_only = FALSE, offset = 1,
  boot = FALSE)

c(fit1$hr, fit1$hr_CI)
```

tssim *Simulate Data for Treatment Switching*

Description

Simulates data for studies involving treatment switching, incorporating time-dependent confounding. The generated data can be used to evaluate methods for handling treatment switching in survival analysis.

Usage

```
tssim(
  tdxo = 0L,
  coxo = 1L,
  allocation1 = 1L,
  allocation2 = 1L,
  p_X_1 = NA_real_,
  p_X_0 = NA_real_,
  rate_T = NA_real_,
  beta1 = NA_real_,
  beta2 = NA_real_,
  gamma0 = NA_real_,
  gamma1 = NA_real_,
  gamma2 = NA_real_,
  gamma3 = NA_real_,
  gamma4 = NA_real_,
  zeta0 = NA_real_,
  zeta1 = NA_real_,
  zeta2 = NA_real_,
  zeta3 = NA_real_,
  alpha0 = NA_real_,
  alpha1 = NA_real_,
  alpha2 = NA_real_,
  theta1_1 = NA_real_,
  theta1_0 = NA_real_,
  theta2 = NA_real_,
  rate_C = NA_real_,
  accrualTime = 0L,
  accrualIntensity = NA_real_,
  followupTime = NA_real_,
  fixedFollowup = 0L,
  plannedTime = NA_real_,
  days = NA_integer_,
  n = NA_integer_,
  NSim = 1000L,
  seed = NA_integer_
)
```

Arguments

`tdxo` Logical indicator for timing of treatment switching:

- 1: Treatment switching can occur at or after disease progression.
- 0: Treatment switching is restricted to the time of disease progression.

`coxo` Logical indicator for arm-specific treatment switching:

- 1: Treatment switching occurs only in the control arm.
- 0: Treatment switching is allowed in both arms.

`allocation1` Number of subjects in the active treatment group in a randomization block. Defaults to 1 for equal randomization.

`allocation2` Number of subjects in the control group in a randomization block. Defaults to 1 for equal randomization.

`p_X_1` Probability of poor baseline prognosis in the experimental arm.

`p_X_0` Probability of poor baseline prognosis in the control arm.

`rate_T` Baseline hazard rate for time to death.

`beta1` Log hazard ratio for randomized treatment (`R`).

`beta2` Log hazard ratio for baseline covariate (`X`).

`gamma0` Intercept for the time-dependent covariate model (`L`).

`gamma1` Coefficient for lagged treatment switching (`Alag`) in the `L` model.

`gamma2` Coefficient for lagged `L` (`Llag`) in the `L` model.

`gamma3` Coefficient for baseline covariate (`X`) in the `L` model.

`gamma4` Coefficient for randomized treatment (`R`) in the `L` model.

`zeta0` Intercept for the disease progression model (`Z`).

`zeta1` Coefficient for `L` in the `Z` model.

`zeta2` Coefficient for baseline covariate (`X`) in the `Z` model.

`zeta3` Coefficient for randomized treatment (`R`) in the `Z` model.

`alpha0` Intercept for the treatment switching model (`A`).

`alpha1` Coefficient for `L` in the `A` model.

`alpha2` Coefficient for baseline covariate (`X`) in the `A` model.

`theta1_1` Negative log time ratio for `A` for the experimental arm.

`theta1_0` Negative log time ratio for `A` for the control arm.

`theta2` Negative log time ratio for `L`.

`rate_C` Hazard rate for random (dropout) censoring.

`accrualTime` A vector that specifies the starting time of piecewise Poisson enrollment time intervals. Must start with 0, e.g., `c(0, 3)` breaks the time axis into 2 accrual intervals: [0, 3) and [3, Inf).

`accrualIntensity` A vector of accrual intensities. One for each accrual time interval.

`followupTime` Follow-up time for a fixed follow-up design.

`fixedFollowup`	Whether a fixed follow-up design is used. Defaults to 0 for variable follow-up.
`plannedTime`	The calendar time for the analysis.
`days`	Number of days in each treatment cycle.
`n`	Number of subjects per simulation.
`NSim`	Number of simulated datasets.
`seed`	Random seed for reproducibility.

Details

The simulation algorithm is adapted from Xu et al. (2022) and simulates disease progression status while incorporating the multiplicative effects of both baseline and time-dependent covariates on survival time. The design options `tdxo` and `coxo` specify the timing of treatment switching and the study arm eligibility for switching, respectively. Time is measured in days.

In a fixed follow-up design, all subjects share the same follow-up duration. In contrast, under a variable follow-up design, follow-up time also depends on each subject's enrollment date. The number of treatment cycles for a subject is determined by dividing the follow-up time by the number of days in each cycle.

1. At randomization, subjects are assigned to treatment arms using block randomization, with `allocation1` patients assigned to active treatment and `allocation2` to control within each randomized block. A baseline covariate is also generated for each subject:

$$X_i \sim \text{Bernoulli}(p_{X_1} R_i + p_{X_0}(1 - R_i))$$

2. The initial survival time is drawn from an exponential distribution with hazard:

$$rate_T \exp(\beta_1 R_i + \beta_2 X_i)$$

   We define the event indicator at cycle j as

$$Y_{i,j} = I(T_i \leq j \times days)$$

3. The initial states are set to $L_{i,0} = 0$, $Z_{i,0} = 0$, $A_{i,0} = 0$, $Y_{i,0} = 0$. For each treatment cycle $j = 1, \ldots, J$, we set $tstart = (j - 1) \times days$.
4. Generate time-dependent covariates:

$$\text{logit} P(L_{i,j} = 1|\text{history}) = \gamma_0 + \gamma_1 A_{i,j-1} + \gamma_2 L_{i,j-1} + \gamma_3 X_i + \gamma_4 R_i$$

5. If $T_i \leq j \times days$, set $tstop = T_i$ and $Y_{i,j} = 1$, which completes data generation for subject i.
6. If $T_i > j \times days$, set $tstop = j \times days$, $Y_{i,j} = 0$, and perform the following before proceeding to the next cycle for the subject.
7. Generate disease progression status: If $Z_{i,j-1} = 0$,

$$\text{logit} P(Z_{i,j} = 1|\text{history}) = \zeta_0 + \zeta_1 L_{i,j} + \zeta_2 X_i + \zeta_3 R_i$$

   Otherwise, set $Z_{i,j} = 1$.

8. Generate alternative therapy status: If $A_{i,j-1} = 0$, determine switching eligibility based on design options. If switching is allowed:

   $$\text{logit}P(A_{i,j} = 1|\text{history}) = \alpha_0 + \alpha_1 L_{i,j} + \alpha_2 X_i$$

   If switching is now allowed, set $A_{i,j} = 0$. If $A_{i,j-1} = 1$, set $A_{i,j} = 1$ (once switched to alternative therapy, remain on alternative therapy).

9. Update survival time based on changes in alternative therapy status and time-dependent covariates:

   $$T_i = j \times days + (T_i - j \times days) \exp\{-(\theta_{1,1} R_i + \theta_{1,0}(1 - R_i))(A_{i,j} - A_{i,j-1}) - \theta_2(L_{i,j} - L_{i,j-1})\}$$

Additional random censoring times are generated from an exponential distribution with hazard rate $rate_C$.

An extra record is generated when the minimum of the latent survival time, the random censoring time, and the administrative censoring time is greater than the number of regular treatment cycles times days per cycle.

Finally we apply the lag function so that $Z_{i,j}$ and $A_{i,j}$ represent the PD status and alternative therapy status at the start of cycle j (and thus remain appplicable for the entire cycle j) for subject i.

To estimate the true treatment effect in a Cox marginal structural model, one can set $\alpha_0 = -\infty$, which effectively forces $A_{i,j} = 0$ (disabling treatment switching). The coefficient for the randomized treatment can then be estimated using a Cox proportional hazards model.

Value

A list of data frames, each containing simulated longitudinal covariate, pd status, alternative therapy status, and event history data with the following variables:

- `id`: Subject identifier.
- `arrivalTime`: The enrollment time for the subject.
- `trtrand`: Randomized treatment assignment (0 = control, 1 = experimental)
- `bprog`: Baseline prognosis (0 = good, 1 = poor).
- `tpoint`: Treatment cycle index.
- `tstart`: Start day of the treatment cycle.
- `tstop`: End day of the treatment cycle.
- `L`: Time-dependent covariate at `tstart` predicting survival and switching; affected by treatment switching.
- `Llag`: Lagged value of `L`.
- `Z`: Disease progression status at `tstart`.
- `A`: Treatment switching status at `tstart`.
- `Alag`: Lagged value of `A`.
- `event`: Death indicator at `tstop`.
- `timeOS`: Observed time to death or censoring.
- `died`: Indicator of death by end of follow-up.

- `progressed`: Indicator of disease progression by end of follow-up.
- `timePD`: Observed time to progression or censoring.
- `xo`: Indicator for whether treatment switching occurred.
- `xotime`: Time of treatment switching (if applicable).
- `censor_time`: Administrative censoring time.

Author(s)

Kaifeng Lu, <kaifenglu@gmail.com>

References

Jessica G. Young, and Eric J. Tchetgen Tchetgen. Simulation from a known Cox MSM using standard parametric models for the g-formula. Statistics in Medicine. 2014;33(6):1001-1014.

NR Latimer, IR White, K Tilling, and U Siebert. Improved two-stage estimation to adjust for treatment switching in randomised trials: g-estimation to address time-dependent confounding. Statistical Methods in Medical Research. 2020;29(10):2900-2918.

Jing Xu, Guohui Liu, and Bingxia Wang. Bias and type I error control in correcting treatment effect for treatment switching using marginal structural models in Phse III oncology trials. Journal of Biopharmaceutical Statistics. 2022;32(6):897-914.

Examples

```
library(dplyr)

simulated.data <- tssim(
  tdxo = 1, coxo = 1, allocation1 = 1, allocation2 = 1,
  p_X_1 = 0.3, p_X_0 = 0.3,
  rate_T = 0.002, beta1 = -0.5, beta2 = 0.3,
  gamma0 = 0.3, gamma1 = -0.9, gamma2 = 0.7, gamma3 = 1.1, gamma4 = -0.8,
  zeta0 = -3.5, zeta1 = 0.5, zeta2 = 0.2, zeta3 = -0.4,
  alpha0 = 0.5, alpha1 = 0.5, alpha2 = 0.4,
  theta1_1 = -0.4, theta1_0 = -0.4, theta2 = 0.2,
  rate_C = 0.0000855, accrualIntensity = 20/30,
  fixedFollowup = FALSE, plannedTime = 1350, days = 30,
  n = 500, NSim = 100, seed = 314159)

simulated.data[[1]] %>% filter(id == 1)
```

Index