Package 'survtmle'

April 16, 2019

Title Compute Targeted Minimum Loss-Based Estimates in Right-Censored Survival Settings

Version 1.1.1

Description Targeted estimates of marginal cumulative incidence in survival settings with and without competing risks, including estimators that respect bounds (Benkeser, Carone, and Gilbert. Statistics in Medicine, 2017. <doi:10.1002/sim.7337>).

Depends R (>= 3.0.0)

Imports Matrix, speedglm, SuperLearner, plyr, dplyr, tidyr (>= 0.8.0), stringr, ggplot2, ggsci

Suggests testthat, knitr, rmarkdown, survival, cmprsk, tibble

License MIT + file LICENSE

URL https://github.com/benkeser/survtmle

BugReports https://github.com/benkeser/survtmle/issues

Encoding UTF-8

LazyData true

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author David Benkeser [aut, cre, cph] (<https://orcid.org/0000-0002-1019-8343>), Nima Hejazi [aut] (<https://orcid.org/0000-0002-7127-2789>)

Maintainer David Benkeser <benkeser@emory.edu>

Repository CRAN

Date/Publication 2019-04-16 15:53:03 UTC

R topics documented:

`checkInputs` *Check Function Inputs*

Description

Check the input values of function parameters for errors.

Usage

```
checkInputs(ftime, ftype, trt, adjustVars, t0 = max(ftime[ftype > 0]),
  SL.ftime = NULL, SL.ctime = NULL, SL.trt = NULL,
  glm.ftime = NULL, glm.ctime = NULL, glm.trt = "1",
  returnIC = TRUE, returnModels = TRUE,
  ftypeOfInterest = unique(ftype[ftype != 0]),
```

```
  trtOfInterest = unique(trt), method = "hazard", bounds = NULL,
  verbose = FALSE, tol = 1/(length(ftime)), maxIter = 100,
  Gcomp = FALSE)
```

Arguments

ftime	A numeric vector of failure times. Right-censored observations should have corresponding ftype set to 0.
ftype	A numeric vector indicating the type of failure. Observations with ftype == 0 are treated as right-censored. Each unique value besides zero is treated as a separate type of failure.
trt	A numeric vector indicating observed treatment assignment. Each unique value will be treated as an (unordered) separate type of treatment. Currently, only two unique values of trt are supported.
adjustVars	A data.frame of adjustment variables that will be used in estimating the conditional treatment, censoring, and failure (hazard or conditional mean) probabilities.
t0	The time at which to return cumulative incidence estimates. By default this is set to max(ftime).
SL.ftime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the outcome regression (either cause-specific hazards or conditional mean). See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
SL.ctime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional hazard for censoring. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
SL.trt	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional probability of treatment. It is expected that the wrappers used in the library will play nicely with the input variables, which will be names(adjustVars).
glm.ftime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression (either cause-specific hazards or conditional mean). Ignored if SL.ftime != NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.ctime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional hazard for censoring. Ignored if SL.ctime != NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.trt	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional probability of treatment. Ignored if SL.trt != NULL. By default set to "1", corresponding

to using empirical estimates of each value of `trt`. The formula can include any variables found in `names(adjustVars)`.

`returnIC` A boolean indicating whether to return vectors of influence curve estimates. These are needed for some post-hoc comparisons, so it is recommended to leave as `TRUE` (the default) unless the user is sure these estimates will not be needed later.

`returnModels` A boolean indicating whether to return the `SuperLearner` or `glm` objects used to estimate the nuisance parameters. Must be set to `TRUE` if the user plans to use calls to `timepoints` to obtain estimates at times other than `t0`. See `?timepoints` for more information.

`ftypeOfInterest`
An input specifying what failure types to compute estimates of incidence for. The default value computes estimates for values `unique(ftype)`. Can alternatively be set to a vector of values found in `ftype`.

`trtOfInterest` An input specifying which levels of `trt` are of interest. The default value computes estimates for values `unique(trt)`. Can alternatively be set to a vector of values found in `trt`.

`method` A character specification of how the targeted minimum loss-based estimators should be computer, either `"mean"` or `"hazard"`. The `"mean"` specification uses a closed-form targeted minimum loss-based estimation based on the G-computation formula of Bang and Robins (2005). The `"hazard"` specification uses an iteratively algorithm based on cause-specific hazard functions. The latter specification has no guarantee of convergence in finite samples. The convergence can be influenced by the stopping criteria specified in the `tol`. Future versions may implement a closed form version of this hazard-based estimator.

`bounds` A list of bounds.

`verbose` A boolean indicating whether the function should print messages to indicate progress.

`tol` The stopping criteria when `method = "hazard"`. The algorithm will continue performing targeting updates to the initial estimators until the empirical mean of the efficient influence function is smaller than `tol`. The default (`1/length(ftime)`) is a sensible value. Larger values can be used in situations where convergence of the algorithm is an issue; however, this may result in large finite-sample bias.

`maxIter` A maximum number of iterations for the algorithm when `method = "hazard"`. The algorithm will iterate until either the empirical mean of the efficient influence function is smaller than `tol` or until `maxIter` iterations have been completed.

`Gcomp` A boolean indicating whether to compute the G-computation estimator (i.e., a substitution estimator with no targeting step). Note, theory does not support inference for the G-computation estimator if Super Learner is used to estimate failure and censoring mechanisms. Only implemented for `method = "mean"`.

Value

Options to be passed to `mean_tmle` or `hazard_tmle`.

cleanglm *Clean up outputs from GLM*

Description

Removes superfluous output from the call to `glm` that is not needed to perform later predictions. It is applied as a space saving technique.

Usage

```
cleanglm(cm)
```

Arguments

`cm`	An object of class `glm` or class `speedglm`.

Value

An object of class `glm` or `speedglm`, but with unnecessary output removed.

confint.survtmle *confint.survtmle*

Description

Computes confidence intervals for a fitted `survtmle` object.

Usage

```
## S3 method for class 'survtmle'
confint(object, parm = seq_along(object$est),
  level = 0.95, ...)
```

Arguments

`object`	An object of class `survtmle`.
`parm`	A numeric vector indicating which indexes of `object$est` to return confidence intervals for (default is to return all).
`level`	The confidence level requested.
`...`	Other arguments. Not currently used.

Value

A matrix with columns giving the lower and upper confidence limits for each parameter. These will be labeled as (1-level)/2 and 1 - (1-level)/2 in percent. The default is 2.5

Examples

```
# simulate data
set.seed(1234)
n <- 100
ftime <- round(runif(n, 1, 4))
ftype <- round(runif(n, 0, 2))
trt <- rbinom(n, 1, 0.5)
adjustVars <- data.frame(W1 = rnorm(n), W2 = rnorm(n))

# fit a survtmle object
fit <- survtmle(ftime = ftime, ftype = ftype, trt = trt,
                adjustVars = adjustVars, glm.trt = "W1 + W2",
                glm.ftime = "trt + W1 + W2", glm.ctime = "trt + W1 + W2",
                method = "mean", t0 = 4)
# get confidence intervals
ci <- confint(fit)
ci
```

`confint.tp.survtmle` *confint.tp.survtmle*

Description

Computes confidence intervals for a fitted `tp.survtmle` object.

Usage

```
## S3 method for class 'tp.survtmle'
confint(object, parm, level = 0.95, ...)
```

Arguments

`object`	An object of class `tp.survtmle`, as produced by invoking the function `timepoints` on an object produced by `survtmle`, for which a confidence interval is to be computed.
`parm`	A numeric vector indicating which indexes of `object$est` to return confidence intervals for (default is to return all). NOT USED NOW.
`level`	A `numeric` indicating the level of the confidence interval to be computed.
`...`	Other arguments. Not currently used.

Value

A list of matrices, each with columns giving the lower and upper confidence limits for each parameter. These will be labeled as (1-level)/2 and 1 - (1-level)/2 in percent. The default is 2.5 contains as many matrices as their are comparison groups in the input data.

Examples

```
# simulate data
set.seed(1234)
n <- 100
ftime <- round(runif(n, 1, 4))
ftype <- round(runif(n, 0, 2))
trt <- rbinom(n, 1, 0.5)
adjustVars <- data.frame(W1 = rnorm(n), W2 = rnorm(n))

# fit a survtmle object
fit <- survtmle(ftime = ftime, ftype = ftype, trt = trt,
                adjustVars = adjustVars, glm.trt = "W1 + W2",
                glm.ftime = "trt + W1 + W2", glm.ctime = "trt + W1 + W2",
                method = "mean", t0 = 4)
# extract cumulative incidence at each timepoint
tpfit <- timepoints(fit, times = seq_len(4))
# get confidence intervals
ci <- confint(tpfit)
ci
```

---

estimateCensoring *Estimate Censoring Mechanisms*

Description

Computes an estimate of the hazard for censoring using either `glm` or `SuperLearner` based on log-likelihood loss. The function then computes the censoring survival distribution based on these estimates. The structure of the function is specific to how it is called within `survtmle`. In particular, `dataList` must have a very specific structure for this function to run properly. The list should consist of `data.frame` objects. The first will have the number of rows for each observation equal to the `ftime` corresponding to that observation. The subsequent entries will have `t0` rows for each observation and will set `trt` column equal to each value of `trtOfInterest` in turn. One of these columns must be named `C` that is a counting process for the right-censoring variable. The function will fit a regression with `C` as the outcome and functions of `trt` and `names(adjustVars)` as specified by `glm.ctime` or `SL.ctime` as predictors.

Usage

```
estimateCensoring(dataList, adjustVars, t0, SL.ctime = NULL,
  glm.ctime = NULL, glm.family, returnModels = FALSE, verbose = TRUE,
  gtol = 0.001, ...)
```

Arguments

`dataList`	A list of `data.frame` objects as described in `?makeDataList`.
`adjustVars`	Object of class `data.frame` that contains the variables to adjust for in the regression.

t0	The timepoint at which survtmle was called to evaluate. Needed only because the naming convention for the regression if t == t0 is different than if t != t0.
SL.ctime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the outcome regression (either cause-specific hazards or conditional mean). See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
glm.ctime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression (either cause-specific hazards or conditional mean). Ignored if SL.ctime != NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.family	The type of regression to be performed if fitting GLMs in the estimation and fluctuation procedures. The default is "binomial" for logistic regression. Only change this from the default if there are justifications that are well understood. This is inherited from the calling function (either mean_tmle or hazard_tmle).
returnModels	A boolean indicating whether to return the SuperLearner or glm objects used to estimate the nuisance parameters. Must be set to TRUE if the user plans to use calls to timepoints to obtain estimates at times other than t0. See ?timepoints for more information.
verbose	A boolean indicating whether the function should print messages to indicate progress.
gtol	The truncation level of predicted censoring survival to handle positivity violations.
...	Other arguments. Not currently used.

Value

The function returns a list that is exactly the same as the input dataList, but with a column named G_dC added to it, which is the estimated conditional survival distribution for the censoring variable evaluated at the each of the rows of each data.frame in dataList.

`estimateHazards` *Estimation for the Method of Cause-Specific Hazards*

Description

This function computes an estimate of the cause-specific hazard functions over all times using either glm or SuperLearner. The structure of the function is specific to how it is called within hazard_tmle. In particular, dataList must have a very specific structure for this function to run properly. The list should consist of data.frame objects. The first will have the number of rows for each observation equal to the ftime corresponding to that observation. The subsequent entries will have t0 rows for each observation and will set trt column equal to each value of trtOfInterest in turn. The function uses the first entry in dataList to iteratively fit hazard regression models for

each cause of failure. Thus, this data.frame needs to have a column called Nj for each value of j in J. The first fit estimates the hazard of min(J), while subsequent fits estimate the pseudo-hazard of all other values of j, where pseudo-hazard is used to mean the probability of a failure due to type j at a particular timepoint given no failure of any type at any previous timepoint AND no failure due to type k < j at a particular timepoint. The hazard estimates of causes j' can then be used to map this pseudo-hazard back into the hazard at a particular time. This is nothing more than the re-framing of a conditional multinomial probability into a series of conditional binomial probabilities. This structure ensures that no strata have estimated hazards that sum to more than one over all possible causes of failure at a particular timepoint.

Usage

```
estimateHazards(dataList, J, adjustVars, SL.ftime = NULL,
  glm.ftime = NULL, glm.family, returnModels, bounds, verbose, ...)
```

Arguments

dataList	A list of data.frame objects.
J	Numeric vector indicating the labels of all causes of failure.
adjustVars	Object of class data.frame that contains the variables to adjust for in the regression.
SL.ftime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the outcome regression (either cause-specific hazards or conditional mean). See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
glm.ftime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression (either using cause-specific hazards or conditional mean). Ignored if SL.ftime != NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.family	The type of regression to be performed if fitting GLMs in the estimation and fluctuation procedures. The default is "binomial" for logistic regression. Only change this from the default if there are justifications that are well understood. This is inherited from the calling function (either mean_tmle or hazard_tmle).
returnModels	A boolean indicating whether to return the SuperLearner or glm objects used to estimate the nuisance parameters. Must be set to TRUE if the user plans to use calls to timepoints to obtain estimates at times other than t0. See ?timepoints for more information.
bounds	A list of bounds... TODO: Add more description here.
verbose	A boolean indicating whether the function should print messages to indicate progress.
...	Other arguments. Not currently used.

Value

The function returns a list that is exactly the same as the input dataList, but with additional columns corresponding to the hazard, pseudo-hazard, and the total hazard for summed over all causes k < j.

---

estimateIteratedMean *Estimation for the Method of Iterated Means*

Description

This function computes an estimate of the G-computation regression at a specified time t using either glm or SuperLearner. The structure of the function is specific to how it is called within mean_tmle. In particular, wideDataList must have a very specific structure for this function to run properly. The list should consist of data.frame objects. The first should have all rows set to their observed value of trt. The remaining should in turn have all rows set to each value of trtOfInterest in the survtmle call. Currently the code requires each data.frame to have named columns for each name in names(adjustVars), as well as a column named trt. It must also have a columns named Nj.Y where j corresponds with the numeric values input in allJ. These are the indicators of failure due to the various causes before time t and are necessary for determining who to include in the regression. Similarly, each data.frame should have a column call C.Y where Y is again t - 1, so that right censored observations are not included in the regressions. The function will fit a regression with Qj.star.t+1 (also needed as a column in wideDataList) on functions of trt and names(adjustVars) as specified by glm.ftime or SL.ftime.

Usage

```
estimateIteratedMean(wideDataList, t, whichJ, allJ, t0, adjustVars,
  SL.ftime = NULL, glm.ftime = NULL, verbose, returnModels = FALSE,
  bounds = NULL, ...)
```

Arguments

wideDataList	A list of data.frame objects.
t	The timepoint at which to compute the iterated mean.
whichJ	Numeric value indicating the cause of failure for which regression should be computed.
allJ	Numeric vector indicating the labels of all causes of failure.
t0	The timepoint at which survtmle was called to evaluate. Needed only because the naming convention for the regression if t == t0 is different than if t != t0.
adjustVars	Object of class data.frame that contains the variables to adjust for in the regression.
SL.ftime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the outcome regression (either cause-specific hazards or conditional mean). See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers

used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).

glm.ftime — A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression (either cause-specific hazards or conditional mean). Ignored if SL.ftime != NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).

verbose — A boolean indicating whether the function should print messages to indicate progress.

returnModels — A boolean indicating whether to return the SuperLearner or glm objects used to estimate the nuisance parameters. Must be set to TRUE if the user plans to use calls to timepoints to obtain estimates at times other than t0. See ?timepoints for more information.

bounds — A list of bounds to be used when performing the outcome regression (Q) with the Super Learner algorithm. NOT YET IMPLEMENTED.

... — Other arguments. Not currently used.

Value

The function then returns a list that is exactly the same as the input wideDataList, but with a column named Qj.t added to it, which is the estimated conditional mean of Qj.star.t+1 evaluated at the each of the rows of each data.frame in wideDataList.

---

estimateTreatment *Estimate Treatment Mechanisms*

Description

This function computes the conditional probability of having trt for each specified level either using glm or SuperLearner. Currently only two unique values of treatment are acceptable. By default the function will compute estimates of the conditional probability of trt == max(trt) and compute the probability of trt == min(trt) as one minus this probability.

Usage

```
estimateTreatment(dat, adjustVars, glm.trt = NULL, SL.trt = NULL,
  returnModels = FALSE, verbose = FALSE, gtol = 0.001, ...)
```

Arguments

dat — An object of class data.frame. Must have named column trt.

adjustVars — An object of class data.frame that will be used either as the data argument in a call to glm or as the X object in a call to SuperLearner.

glm.trt	A character formula for the right-hand side of formula in a call to glm. See ?survtmle for more documentation. Alternatively, this could be an object of class glm (as in calls to this function via timepoints), in which case predictions are obtained using this object with no new fitting.
SL.trt	A valid specification of the SL.library option of a call to SuperLearner. See ?survtmle for more documentation. Alternatively, this could be an object of class SuperLearner (as in calls to this function via timepoints), in which case predictions are obtained using this object with no new fitting.
returnModels	A boolean indicating whether fitted model objects should be returned.
verbose	A boolean passed to the verbose option of the call to SuperLearner.
gtol	The truncation level of predicted trt probabilities to handle positivity violations.
...	Other arguments. Not currently used

Value

dat The input data.frame object with two added columns corresponding with the conditional probability (given adjustVars) of trt==max(trt) and trt==min(trt).

trtMod If returnModels = TRUE, the fitted glm or SuperLearner object. Otherwise, NULL

---

fast_glm *Wrapper for faster Generalized Linear Models*

Description

A convenience utility to fit regression models more quickly in the main internal functions for estimation, which usually require logistic regression. Use of speedglm appears to provide roughly an order of magnitude improvement in speed when compared to glm in custom benchmarks.

Usage

```
fast_glm(reg_form, data, family, ...)
```

Arguments

reg_form	Object of class formula indicating the regression to be fit.
data	Object of class data.frame containing the data.
family	Object of class family from package stats indicating the error distribution. Appropriate options are limited to gaussian and binomial.
...	Additional arguments passed to glm or speedglm.

Value

Object of class glm or speedglm.

`fluctuateHazards` *Fluctuation for the Method of Cause-Specific Hazards*

Description

This function performs a fluctuation of an initial estimate of the cause-specific hazard functions using a call to `glm` (i.e., a logistic submodel) or a call to `optim` (to ensure fluctuations stay within model space). The structure of the function is specific to how it is called within `hazard_tmle`. In particular, `dataList` must have a very specific structure for this function to run properly. The list should consist of `data.frame` objects. The first will have the number of rows for each observation equal to the `ftime` corresponding to that observation. The subsequent entries will have `t0` rows for each observation and will set `trt` column equal to each value of `trtOfInterest` in turn. The function will fit a logistic regression with (a scaled version of) `Nj` as outcome, the logit of the current (pseudo-) hazard estimate as offset and the targeted minimum loss-based estimation "clever covariates". The function then obtains predictions based on this fit on each of the `data.frame` objects in `dataList`.

Usage

```
fluctuateHazards(dataList, allJ, ofInterestJ, nJ, uniqtrt, ntrt, t0,
  verbose, ...)
```

Arguments

`dataList`	A list of `data.frame` objects.
`allJ`	Numeric vector indicating the labels of all causes of failure.
`ofInterestJ`	Numeric vector indicating `ftypeOfInterest` that was passed to `hazard_tmle`.
`nJ`	The number of unique failure types.
`uniqtrt`	The values of `trtOfInterest` passed to `mean_tmle`.
`ntrt`	The number of `trt` values of interest.
`t0`	The timepoint at which `survtmle` was called to evaluate.
`verbose`	A boolean indicating whether the function should print messages to indicate progress.
`...`	Other arguments. Not currently used.

Value

The function returns a list that is exactly the same as the input `dataList`, but with updated columns corresponding with estimated cumulative incidence at each time and estimated "clever covariates" at each time.

`fluctuateIteratedMean` *Fluctuation for the Method of Iterated Means*

Description

This function performs a fluctuation of an initial estimate of the G-computation regression at a specified time `t` using a call to `glm` (i.e., a logistic submodel) or a call to `optim` (if bounds are specified). The structure of the function is specific to how it is called within `mean_tmle`. In particular, `wideDataList` must have a very specific structure for this function to run properly. The list should consist of `data.frame` objects. The first should have all rows set to their observed value of `trt`. The remaining should in turn have all rows set to each value of `trtOfInterest` in the `survtmle` call. The latter will be used to obtain predictions that are then mapped into the estimates of the cumulative incidence function at `t0`. Currently the code requires each `data.frame` to have named columns for each name in `names(adjustVars)`, as well as a column named `trt`. It must also have a columns named `Nj.Y` where j corresponds with the numeric values input in `allJ`. These are the indicators of failure due to the various causes before time `t` and are necessary for determining who to include in the fluctuation regression. Similarly, each `data.frame` should have a column call `C.Y` where Y is again `t-1`, so that right censored observations are not included in the regressions. The function will fit a logistic regression with `Qj.star.t + 1` as outcome (also needed as a column in `wideDataList`) with offset `qlogis(Qj.star.t)` and number of additional covariates given by `length(trtOfInterest)`. These additional covariates should be columns in the each `data.frame` in `wideDataList` called `H.z.t` where `z` corresponds to a each unique value of `trtOfInterest`. The function returns the same `wideDataList`, but with a column called `Qj.star.t` added to it, which is the fluctuated initial regression estimate evaluated at the observed data points.

Usage

```
fluctuateIteratedMean(wideDataList, t, uniqtrt, whichJ, allJ, t0,
  Gcomp = FALSE, bounds = NULL, ...)
```

Arguments

`wideDataList`	A list of `data.frame` objects.
`t`	The timepoint at which to compute the iterated mean.
`uniqtrt`	The values of `trtOfInterest` passed to `mean_tmle`.
`whichJ`	Numeric value indicating the cause of failure for which regression should be computed.
`allJ`	Numeric vector indicating the labels of all causes of failure.
`t0`	The timepoint at which `survtmle` was called to evaluate. Needed only because the naming convention for the regression if `t == t0` is different than if `t != t0`.
`Gcomp`	A boolean indicating whether `mean_tmle` was called to evaluate the G-computation estimator, in which case this function does nothing but re-label columns.
`bounds`	A list of bounds to be used when performing the outcome regression (Q) with the Super Learner algorithm. NOT YET IMPLEMENTED.
`...`	Other arguments. Not currently used.

Value

The function then returns a list that is exactly the same as the input `wideDataList`, but with a column named `Qj.star.t` added to it, which is the fluctuated conditional mean of `Qj.star.t+1` evaluated at the each of the rows of each `data.frame` in `wideDataList`.

`format.perc` *format.perc*

Description

Copied from package `stats`.

Usage

```
## S3 method for class 'perc'
format(probs, digits)
```

Arguments

`probs`	Probabilities
`digits`	Number of digits to round to

`getHazardInfluenceCurve`
Extract Influence Curve for Estimated Hazard Functions

Description

This function computes the hazard-based efficient influence curve at the final estimate of the fluctuated cause-specific hazard functions and evaluates it on the observed data. The influence-function is computed on the long-format data but is subsequently summed over all timepoints for each observation and the function returns a new short form data set with columns added corresponding to the sum over all timepoints of the estimated efficient influence function evaluated at that observation.

Usage

```
getHazardInfluenceCurve(dataList, dat, allJ, ofInterestJ, nJ, uniqtrt, t0,
  verbose, ...)
```

Arguments

dataList	A list of data.frame objects. See ?makeDataList for more information.
dat	A data.frame in short form. See ?makeDataList for more information.
allJ	Numeric vector indicating the labels of all causes of failure.
ofInterestJ	Numeric vector indicating ftypeOfInterest that was passed to hazard_tmle.
nJ	The number of unique failure types.
uniqtrt	The values of trtOfInterest passed to mean_tmle.
t0	The timepoint at which survtmle was called to evaluate.
verbose	A boolean indicating whether the function should print messages to indicate progress.
...	Other arguments. Not currently used.

Value

An object of class data.frame with columns D.jX.zZ added for each value of X in ofInterestJ and each value of Z in uniqtrt. These are the sum over all timepoints of the estimated efficient influence function evaluated at that observation.

---

grad *Gradient for Logistic Regression*

Description

A function that computes the gradient of the for a logistic regression model. Used by optim on occasion.

Usage

```
grad(beta, Y, X)
```

Arguments

beta	A vector of coefficients in a logistic glm
Y	The outcome
X	The design matrix

Value

Numeric vector of the gradient of the parameter vector

grad_offset *Gradient for Logistic Regression with Offsets*

Description

A function that computes the gradient of the for a logistic regression model with an offset term. Used by `optim` on occasion.

Usage

```
grad_offset(beta, Y, H, offset = NULL)
```

Arguments

`beta`	A vector of coefficients in a logistic glm
`Y`	The outcome
`H`	The covariate matrix
`offset`	The offset vector

Value

Numeric vector of the gradient of the parameter vector

hazard_tmle *TMLE for Cause-Specific Hazard Functions*

Description

This function estimates the marginal cumulative incidence for failures of specified types using targeted minimum loss-based estimation based on the initial estimates of the cause-specific hazard functions for failures of each type. The function is called by `survtmle` whenever `method = "hazard"` is specified. However, power users could, in theory, make calls directly to this function.

Usage

```
hazard_tmle(ftime, ftype, trt, t0 = max(ftime[ftype > 0]),
  adjustVars = NULL, SL.ftime = NULL, SL.ctime = NULL,
  SL.trt = NULL, glm.ftime = NULL, glm.ctime = NULL, glm.trt = "1",
  glm.family = "binomial", returnIC = TRUE, returnModels = FALSE,
  ftypeOfInterest = unique(ftype[ftype != 0]),
  trtOfInterest = unique(trt), bounds = NULL, verbose = FALSE,
  tol = 1/(length(ftime)), maxIter = 100, gtol = 0.001, ...)
```

Arguments

ftime	A numeric vector of failure times. Right-censored observations should have corresponding ftype set to 0.
ftype	A numeric vector indicating the type of failure. Observations with ftype=0 are treated as a right-censored observation. Each unique value besides zero is treated as a separate type of failure.
trt	A numeric vector indicating observed treatment assignment. Each unique value will be treated as a different type of treatment. Currently, only two unique values are supported.
t0	The time at which to return cumulative incidence estimates. By default this is set to max(ftime[ftype > 0]).
adjustVars	A data.frame of adjustment variables that will be used in estimating the conditional treatment, censoring, and failure (hazard or conditional mean) probabilities.
SL.ftime	A character vector or list specification to be passed to the SL.library option in the call to SuperLearner for the cause-specific hazards. See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt", names(adjustVars), and "t" if method = "hazard".
SL.ctime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional hazard for censoring. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
SL.trt	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional probability of treatment. It is expected that the wrappers used in the library will play nicely with the input variables, which will be names(adjustVars).
glm.ftime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression. Ignored if SL.ftime is not equal to NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.ctime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional hazard for censoring. Ignored if SL.ctime is not equal to NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.trt	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional probability of treatment. Ignored if SL.trt is not equal to NULL. The formula can include any variables found in names(adjustVars).
glm.family	The type of regression to be performed if fitting GLMs in the estimation and fluctuation procedures. The default is "binomial" for logistic regression. Only change this from the default if there are justifications that are well understood. This is passed directly to estimateCensoring and estimateHazards.

`returnIC` A boolean indicating whether to return vectors of influence curve estimates. These are needed for some post-hoc comparisons, so it is recommended to leave as `TRUE` (the default) unless the user is sure these estimates will not be needed later.

`returnModels` A boolean indicating whether to return the `SuperLearner` or `glm` objects used to estimate the nuisance parameters. Must be set to `TRUE` if the user plans to use `timepoints` to obtain estimates of incidence at times other than `t0`. See `?timepoints` for more information.

`ftypeOfInterest` An input specifying what failure types to compute estimates of incidence for. The default value computes estimates for values `unique(ftype)`. Can alternatively be set to a vector of values found in `ftype`.

`trtOfInterest` An input specifying which levels of `trt` are of interest. The default value computes estimates for values `unique(trt)`. Can alternatively be set to a vector of values found in `trt`.

`bounds` A `data.frame` of bounds on the conditional hazard function. The `data.frame` should have a column named `"t"` that includes values `seq_len(t0)`. The other columns should be names `paste0("l",j)` and `paste0("u",j)` for each unique failure type label j, denoting lower and upper bounds, respectively. See examples.

`verbose` A boolean indicating whether the function should print messages to indicate progress. If `SuperLearner` is called internally, this option will additionally be passed to `SuperLearner`.

`tol` The stopping criteria. The TMLE algorithm performs updates to the initial estimators until the empirical mean of the efficient influence function is smaller than `tol` or until `maxIter` iterations have been completed. The default (`1 / length(ftime)`) is a sensible value. Larger values can be used in situations where convergence of the algorithm is an issue; however, this may result in large finite-sample bias.

`maxIter` The maximum number of iterations for the algorithm. The algorithm will iterate until either the empirical mean of the efficient influence function is smaller than `tol` or until `maxIter` iterations have been completed.

`gtol` The truncation level of predicted censoring survival. Setting to larger values can help performance in data sets with practical positivity violations.

`...` Other options. Not currently used.

Value

An object of class `survtmle`.

call The call to `survtmle`.

est A numeric vector of point estimates – one for each combination of `ftypeOfInterest` and `trtOfInterest`.

var A covariance matrix for the point estimates.

meanIC The empirical mean of the efficient influence function at the estimated, targeted nuisance parameters. Each value should be small or the user will be warned that excessive finite-sample bias may exist in the point estimates.

ic The efficient influence function at the estimated, fluctuated nuisance parameters, evaluated on each of the observations. These are used to construct confidence intervals for post-hoc comparisons.

ftimeMod If `returnModels = TRUE` the fit object(s) for the call to `glm` or `SuperLearner` for the outcome regression models. If `method="mean"` this will be a list of length `length(ftypeOfInterest)` each of length `t0` (one regression for each failure type and for each timepoint). If `method = "hazard"` this will be a list of length `length(ftypeOfInterest)` with one fit corresponding to the hazard for each cause of failure. If `returnModels = FALSE`, this entry will be `NULL`.

ctimeMod If `returnModels = TRUE` the fit object for the call to `glm` or `SuperLearner` for the pooled hazard regression model for the censoring distribution. If `returnModels = FALSE`, this entry will be `NULL`.

trtMod If `returnModels = TRUE` the fit object for the call to `glm` or `SuperLearner` for the conditional probability of `trt` regression model. If `returnModels = FALSE`, this entry will be `NULL`.

t0 The timepoint at which the function was evaluated.

ftime The numeric vector of failure times used in the fit.

ftype The numeric vector of failure types used in the fit.

trt The numeric vector of treatment assignments used in the fit.

adjustVars The data.frame of failure times used in the fit.

Examples

```
## Single failure type examples
# simulate data
set.seed(1234)
n <- 100
trt <- rbinom(n, 1, 0.5)
adjustVars <- data.frame(W1 = round(runif(n)), W2 = round(runif(n, 0, 2)))

ftime <- round(1 + runif(n, 1, 4) - trt + adjustVars$W1 + adjustVars$W2)
ftype <- round(runif(n, 0, 1))

# Fit 1 - fit hazard_tmle object with GLMs for treatment, censoring, failure
fit1 <- hazard_tmle(ftime = ftime, ftype = ftype,
                    trt = trt, adjustVars = adjustVars,
                    glm.trt = "W1 + W2",
                    glm.ftime = "trt + W1 + W2",
                    glm.ctime = "trt + W1 + W2",
                    returnModels = TRUE)
```

LogLikelihood *Log-Likelihood*

Description

Computes the log-likelihood for a model. Used by `optim` on occasion.

Usage

```
LogLikelihood(beta, X, Y)
```

Arguments

`beta`	A vector of coefficients in a logistic GLM.
`X`	The design matrix.
`Y`	The outcome.

Value

Numeric of the summed negative log-likelihood loss over observations.

LogLikelihood_offset *Log-Likelihood Offset*

Description

Computes the log-likelihood for a logistic regression model with an offset. Used by `optim` on occasion.

Usage

```
LogLikelihood_offset(beta, Y, H, offset)
```

Arguments

`beta`	A vector of coefficients in a logistic GLM.
`Y`	A vector of the outcome.
`H`	The matrix of covariates.
`offset`	The vector of offsets.

Value

Numeric of the summed negative log-likelihood loss over observations.

makeDataList *Convert Short Form Data to List of Wide Form Data*

Description

The function takes a `data.frame` of short format right-censored failure times and reshapes the long format into the wide format needed for calls to both `mean_tmle` and `hazard_tmle`. The list returned will have a number of entries equal to `length(trtOfInterest) + 1`. The first will have number of rows for each observation equal to the `ftime` corresponding to that observation. The subsequent entries will have `t0` rows for each observation and will set `trt` column equal to each value of `trtOfInterest` in turn.

Usage

```
makeDataList(dat, J, ntrt, uniqtrt, t0, bounds = NULL, ...)
```

Arguments

`dat`	The short form `data.frame`
`J`	The unique values of `ftype` passed to `survtmle`.
`ntrt`	The number of `trt` values of interest.
`uniqtrt`	The unique values of `trtOfInterest` passed to `mean_tmle`.
`t0`	The timepoint at which `survtmle` was called to evaluate.
`bounds`	Minimum and maximum values to be placed on the `ftype`.
`...`	Other arguments. Not currently used.

Value

A list of `data.frame` objects as described above.

makeWideDataList *Convert Long Form Data to List of Wide Form Data*

Description

The function takes a `data.frame` and `list` consisting of short and long format right-censored failure times. The function reshapes the long format into the wide format needed for calls to `mean_tmle`. The list returned by the function will have number of entries equal to `length(trtOfInterest) + 1`. The first will contain the observed `trt` columns and will set `C.t` (the censoring counting process) equal to the observed value of censoring. The subsequent entries will set `trt` equal to each level of `trtOfInterest` and set `C.t` to zero for everyone.

Usage

```
makeWideDataList(dat, allJ, uniqtrt, adjustVars, dataList, t0, ...)
```

Arguments

dat	The short form data.frame
allJ	Numeric vector indicating the labels of all causes of failure.
uniqtrt	The values of trtOfInterest passed to mean_tmle.
adjustVars	A data.frame of adjustment variables that will be used in estimating the conditional treatment, censoring, and failure (hazard or conditional mean) probabilities.
dataList	A list of long format data.frame objects. See ?makeDataList for more details on formatting.
t0	The timepoint at which survtmle was called to evaluate.
...	Other arguments. Not currently used.

Value

A list of data.frame objects as described above.

---

mean_tmle *TMLE for G-Computation of Cumulative Incidence*

Description

This function estimates the marginal cumulative incidence for failures of specified types using targeted minimum loss-based estimation based on the G-computation representation of cumulative incidence. The function is called by survtmle whenever method = "mean" is specified. However, power users could, in theory, make calls directly to this function.

Usage

```
mean_tmle(ftime, ftype, trt, t0 = max(ftime[ftype > 0]),
  adjustVars = NULL, SL.ftime = NULL, SL.ctime = NULL,
  SL.trt = NULL, glm.ftime = NULL, glm.ctime = NULL, glm.trt = "1",
  glm.family = "binomial", returnIC = TRUE, returnModels = FALSE,
  ftypeOfInterest = unique(ftype[ftype != 0]),
  trtOfInterest = unique(trt), bounds = NULL, verbose = FALSE,
  Gcomp = FALSE, gtol = 0.001, ...)
```

Arguments

ftime	A numeric vector of failure times. Right-censored observations should have corresponding ftype set to 0.
ftype	A numeric vector indicating the type of failure. Observations with ftype=0 are treated as a right-censored observation. Each unique value besides zero is treated as a separate type of failure.
trt	A numeric vector indicating observed treatment assignment. Each unique value will be treated as a different type of treatment. Currently, only two unique values are supported.
t0	The time at which to return cumulative incidence estimates. By default this is set to max(ftime[ftype > 0]).
adjustVars	A data.frame of adjustment variables that will be used in estimating the conditional treatment, censoring, and failure (hazard or conditional mean) probabilities.
SL.ftime	A character vector or list specification to be passed to the SL.library option in the call to SuperLearner for the outcome regression (either cause-specific hazards or iterated mean). See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt", names(adjustVars), and "t" (if method = "hazard").
SL.ctime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional hazard for censoring. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
SL.trt	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional probability of treatment. It is expected that the wrappers used in the library will play nicely with the input variables, which will be names(adjustVars).
glm.ftime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression. Ignored if SL.ftime is not equal to NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.ctime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional hazard for censoring. Ignored if SL.ctime is not equal to NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.trt	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional probability of treatment. Ignored if SL.trt is not equal to NULL. The formula can include any variables found in names(adjustVars).
glm.family	The type of regression to be performed if fitting GLMs in the estimation and fluctuation procedures. The default is "binomial" for logistic regression. Only change this from the default if there are justifications that are well understood. This is passed directly to estimateCensoring.

`returnIC` A boolean indicating whether to return vectors of influence curve estimates. These are needed for some post-hoc comparisons, so it is recommended to leave as `TRUE` (the default) unless the user is sure these estimates will not be needed later.

`returnModels` A boolean indicating whether to return the `SuperLearner` or `glm` objects used to estimate the nuisance parameters. Must be set to `TRUE` if the user plans to use `timepoints` to obtain estimates of incidence at times other than `t0`. See `?timepoints` for more information.

`ftypeOfInterest` An input specifying what failure types to compute estimates of incidence for. The default value computes estimates for values `unique(ftype)`. Can alternatively be set to a vector of values found in `ftype`.

`trtOfInterest` An input specifying which levels of `trt` are of interest. The default value computes estimates for values `unique(trt)`. Can alternatively be set to a vector of values found in `trt`.

`bounds` A `data.frame` of bounds on the conditional hazard function (if `method = "hazard"`) or on the iterated conditional means (if `method = "mean"`). The `data.frame` should have a column named `"t"` that includes values `1:t0`. The other columns should be names `paste0("l",j)` and `paste0("u",j)` for each unique failure type label j, denoting lower and upper bounds, respectively. See examples.

`verbose` A boolean indicating whether the function should print messages to indicate progress. If `SuperLearner` is called internally, this option will additionally be passed to `SuperLearner`.

`Gcomp` A boolean indicating whether to compute the G-computation estimator (i.e., a substitution estimator with no targeting step). Theory does not support inference for the G-computation estimator if Super Learner is used to estimate failure and censoring distributions. The G-computation is only implemented if `method = "mean"`.

`gtol` The truncation level of predicted censoring survival. Setting to larger values can help performance in data sets with practical positivity violations.

`...` Other options. Not currently used.

Value

An object of class `survtmle`.

call The call to `survtmle`.

est A numeric vector of point estimates – one for each combination of `ftypeOfInterest` and `trtOfInterest`.

var A covariance matrix for the point estimates.

meanIC The empirical mean of the efficient influence function at the estimated, targeted nuisance parameters. Each value should be small or the user will be warned that excessive finite-sample bias may exist in the point estimates.

ic The efficient influence function at the estimated, fluctuated nuisance parameters, evaluated on each of the observations. These are used to construct confidence intervals for post-hoc comparisons.

ftimeMod If `returnModels=TRUE` the fit object(s) for the call to `glm` or `SuperLearner` for the outcome regression models. If `method="mean"` this will be a list of length `length(ftypeOfInterest)` each of length `t0` (one regression for each failure type and for each timepoint). If `method="hazard"` this will be a list of length `length(ftypeOfInterest)` with one fit corresponding to the hazard for each cause of failure. If `returnModels = FALSE`, this entry will be `NULL`.

ctimeMod If `returnModels = TRUE` the fit object for the call to `glm` or `SuperLearner` for the pooled hazard regression model for the censoring distribution. If `returnModels = FALSE`, this entry will be `NULL`.

trtMod If `returnModels = TRUE` the fit object for the call to `glm` or `SuperLearner` for the conditional probability of `trt` regression model. If `returnModels = FALSE`, this entry will be `NULL`.

t0 The timepoint at which the function was evaluated.

ftime The numeric vector of failure times used in the fit.

ftype The numeric vector of failure types used in the fit.

trt The numeric vector of treatment assignments used in the fit.

adjustVars The `data.frame` of failure times used in the fit.

Examples

```
## Single failure type examples
# simulate data
set.seed(1234)
n <- 100
trt <- rbinom(n,1,0.5)
adjustVars <- data.frame(W1 = round(runif(n)), W2 = round(runif(n, 0, 2)))

ftime <- round(1 + runif(n, 1, 4) - trt + adjustVars$W1 + adjustVars$W2)
ftype <- round(runif(n, 0, 1))

# Fit 1 - fit mean_tmle object with GLMs for treatment, censoring, failure
fit1 <- mean_tmle(ftime = ftime, ftype = ftype,
                  trt = trt, adjustVars = adjustVars,
                  glm.trt = "W1 + W2",
                  glm.ftime = "trt + W1 + W2",
                  glm.ctime = "trt + W1 + W2")
```

---

plot.tp.survtmle *Plot Results of Cumulative Incidence Estimates*

Description

Step function plots for both raw and smoothed (monotonic) estimates, the latter by isotonic regression of the raw estimates, of cumulative incidence.

Usage

```
## S3 method for class 'tp.survtmle'
plot(x, ..., type = c("iso", "raw"),
  pal = ggsci::scale_color_lancet())
```

Arguments

x	object of class tp.survtmle as produced by a sequence of appropriate calls to survtmle and timepoints
...	additional arguments passed plot as necessary
type	character describing whether to provide a plot of raw ("raw") or monotonic ("iso") estimates in the resultant step function plot, with the latter being computed by a call to stats::isoreg
pal	A ggplot2 palette object from the ggsci package. The default of scale_color_lancet is generally appropriate for medical and epidemiologic applications, though there are situations in which one might opt to change this. Note that this can also be overridden in the resultant plot object using standard ggplot2 semantics.

Value

object of class ggplot containing a step function plot of the raw or smoothened point estimates of cumulative incidence across a series of timepoints of interest.

Examples

```
library(survtmle)
set.seed(341796)
n <- 100
t_0 <- 10
W <- data.frame(W1 = runif(n), W2 = rbinom(n, 1, 0.5))
A <- rbinom(n, 1, 0.5)
T <- rgeom(n,plogis(-4 + W$W1 * W$W2 - A)) + 1
C <- rgeom(n, plogis(-6 + W$W1)) + 1
ftime <- pmin(T, C)
ftype <- as.numeric(ftime == T)
suppressWarnings(
  fit <- survtmle(ftime = ftime, ftype = ftype,
                  adjustVars = W, glm.ftime = "I(W1*W2) + trt + t",
                  trt = A, glm.ctime = "W1 + t", method = "hazard",
                  verbose = TRUE,  t0 = t_0, maxIter = 2)
)
tpfit <- timepoints(fit, times = seq_len(t_0))
plot(tpfit)
```

print.survtmle *print.survtmle*

Description

The print method for an object of class survtmle

Usage

```
## S3 method for class 'survtmle'
print(x, ...)
```

Arguments

x	An object of class survtmle
...	Other options (not currently used)

Value

Prints the estimates of cumulative incidence and the diagonal of the estimated covariance matrix.

print.tp.survtmle *print.tp.survtmle*

Description

The print method for a timepoints object of class tp.survtmle

Usage

```
## S3 method for class 'tp.survtmle'
print(x, ...)
```

Arguments

x	An object of class tp.survtmle.
...	Other options (not currently used).

Value

Prints the estimates of cumulative incidence and the diagonal of the estimated covariance matrix.

`rtss` *Mock RTSS/AS01 data set*

Description

A dataset containing data that is similar in structure to the RTSS/AS01 malaria vaccine trial. Privacy agreements prevent the sharing of the real data, so please note THAT THIS IS NOT THE REAL RTSS,S DATA. The data set is a `list` of 10 simulated multiple outputation draws. The covariate data, `ftime`, and `vaccine` stay the same across the data sets; however, the `ftype` variable changes, simulating output data sets of multiply infected trial participants.

Usage

```
rtss
```

Format

A `list` with 10 entries, each a `data.frame` with 6,890 rows and 18 columns.

ftime number of months until first recorded malaria disease

ftype the genotype of sampled malaria parasite (0 = censored, 1 = CSP matched, 2 = CSP mismatched)

vaccine vaccine assignment (0 = control vaccine, 1 = vaccine)

ageWeeks participant's age in weeks at trial enrollment

weightForAgeZscore WHO weight-for-age Z-score

sex participant's sex (0 = male, 1 = female)

site1-5 Indicator of study site

heightForAgeZscore WHO height-for-age Z-score

weightForHeightZscore WHO weight-for-height Z-score

armCircumZscore WHO arm circumference Z-score

hemog hemoglobin

distInpatient distance from nearest inpatient clinic

distOutpatient distance from nearest outpatient clinic

startMonthCat study site-specific indicator of rainy (=1) versus dry (=0) season ...

rv144 *Mock RV144 data set*

Description

A dataset containing data that is similar in structure to the RV144 "Thai trial" of the ALVAC/AIDSVAX vaccine. Privacy agreements prevent the sharing of the real data, so please note THAT THIS IS NOT THE REAL RV144 DATA.

Usage

```
rv144
```

Format

A data frame with 15,955 rows and 10 columns:

ftime number of six month visit windows until first recorded incidence of HIV

ftype the genotype of HIV (0 = censored, 1 = amino acid site 169 matched, 2 = amino acid site 169 mismatched)

vax vaccine assignment (0 = placebo, 1 = vaccine)

male male gender (0 = no, 1 = yes)

year04 trial enrollment year 2004 (0 = no, 1 = yes)

year05 trial enrollment year 2005 (0 = no, 1 = yes)

medRisk medium category of risk behaviors (0 = no, 1 = yes)

highRisk high category of risk behaviors (0 = no, 1 = yes)

medAge medium category for age (0 = no, 1 = yes)

highAge high category for age (0 = no, 1 = yes) ...

survtmle *Compute Targeted Minimum Loss-Based Estimators in Survival Analysis Settings*

Description

This function estimates the marginal cumulative incidence for failures of specified types using targeted minimum loss-based estimation.

Usage

```
survtmle(ftime, ftype, trt, adjustVars, t0 = max(ftime[ftype > 0]),
  SL.ftime = NULL, SL.ctime = NULL, SL.trt = NULL,
  glm.ftime = NULL, glm.ctime = NULL, glm.trt = NULL,
  returnIC = TRUE, returnModels = TRUE,
  ftypeOfInterest = unique(ftype[ftype != 0]),
  trtOfInterest = unique(trt), method = "hazard", bounds = NULL,
  verbose = FALSE, tol = 1/(sqrt(length(ftime))), maxIter = 10,
  Gcomp = FALSE, gtol = 0.001)
```

Arguments

ftime	An integer-valued vector of failure times. Right-censored observations should have corresponding ftype set to 0.
ftype	An integer-valued vector indicating the type of failure. Observations with ftype=0 are treated as a right-censored observation. Each unique value besides zero is treated as a separate type of failure.
trt	A numeric vector indicating observed treatment assignment. Each unique value will be treated as a different type of treatment. Currently, only two unique values are supported.
adjustVars	A data.frame of adjustment variables that will be used in estimating the conditional treatment, censoring, and failure (hazard or conditional mean) probabilities.
t0	The time at which to return cumulative incidence estimates. By default this is set to max(ftime[ftype > 0]).
SL.ftime	A character vector or list specification to be passed to the SL.library option in the call to SuperLearner for the outcome regression (either cause-specific hazards or iterated mean). See ?SuperLearner for more information on how to specify valid SuperLearner libraries. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt", names(adjustVars), and "t" (if method="hazard").
SL.ctime	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional hazard for censoring. It is expected that the wrappers used in the library will play nicely with the input variables, which will be called "trt" and names(adjustVars).
SL.trt	A character vector or list specification to be passed to the SL.library argument in the call to SuperLearner for the estimate of the conditional probability of treatment. It is expected that the wrappers used in the library will play nicely with the input variables, which will be names(adjustVars).
glm.ftime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the outcome regression. Ignored if SL.ftime is not equal to NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).
glm.ctime	A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional hazard for

censoring. Ignored if SL.ctime is not equal to NULL. Use "trt" to specify the treatment in this formula (see examples). The formula can additionally include any variables found in names(adjustVars).

glm.trt — A character specification of the right-hand side of the equation passed to the formula option of a call to glm for the estimate of the conditional probability of treatment. Ignored if SL.trt is not equal to NULL. The formula can include any variables found in names(adjustVars).

returnIC — A boolean indicating whether to return vectors of influence curve estimates. These are needed for some post-hoc comparisons, so it is recommended to leave as TRUE (the default) unless the user is sure these estimates will not be needed later.

returnModels — A boolean indicating whether to return the SuperLearner or glm objects used to estimate the nuisance parameters. Must be set to TRUE if the user plans to use timepoints to obtain estimates of incidence at times other than t0. See ?timepoints for more information.

ftypeOfInterest — An input specifying what failure types to compute estimates of incidence for. The default value computes estimates for values unique(ftype). Can alternatively be set to a vector of values found in ftype.

trtOfInterest — An input specifying which levels of trt are of interest. The default value computes estimates for values unique(trt). Can alternatively be set to a vector of values found in trt.

method — A character specification of how the targeted minimum loss-based estimators should be computed, either "mean" or "hazard". The "mean" specification uses a closed-form targeted minimum loss-based estimation based on the G-computation formula of Bang and Robins (2005). The "hazard" specification uses an iteratively algorithm based on cause-specific hazard functions. The latter specification has no guarantee of convergence in finite samples. The convergence can be influenced by the stopping criteria specified in the tol. Future versions may implement a closed form version of this hazard-based estimator.

bounds — A data.frame of bounds on the conditional hazard function (if method = "hazard") or on the iterated conditional means (if method = "mean"). The data.frame should have a column named "t" that includes values seq_len(t0). The other columns should be names paste0("l",j) and paste0("u",j) for each unique failure type label j, denoting lower and upper bounds, respectively. See examples.

verbose — A boolean indicating whether the function should print messages to indicate progress. If SuperLearner is called internally, this option will additionally be passed to SuperLearner.

tol — The stopping criteria when method="hazard". The TMLE algorithm performs updates to the initial estimators until the empirical mean of the efficient influence function is smaller than tol or until maxIter iterations have been completed. The default (1/length(ftime)) is a sensible value. Larger values can be used in situations where convergence of the algorithm is an issue; however, this may result in large finite-sample bias.

`maxIter` A maximum number of iterations for the algorithm when `method = "hazard"`. The algorithm will iterate until either the empirical mean of the efficient influence function is smaller than `tol` or until `maxIter` iterations have been completed.

`Gcomp` A boolean indicating whether to compute the G-computation estimator (i.e., a substitution estimator with no targeting step). Theory does not support inference for the Gcomp estimator if Super Learner is used to estimate failure and censoring distributions. The G-computation is only implemented for `method = "mean"`.

`gtol` The truncation level of predicted censoring survival. Setting to larger values can help performance in data sets with practical positivity violations.

Value

An object of class `survtmle`.

call The call to `survtmle`.

est A numeric vector of point estimates – one for each combination of `ftypeOfInterest` and `trtOfInterest`.

var A covariance matrix for the point estimates.

meanIC The empirical mean of the efficient influence function at the estimated, targeted nuisance parameters. Each value should be small or the user will be warned that excessive finite-sample bias may exist in the point estimates.

ic The efficient influence function at the estimated, fluctuated nuisance parameters, evaluated on each of the observations. These are used to construct confidence intervals for post-hoc comparisons.

ftimeMod If `returnModels=TRUE` the fit object(s) for the call to `glm` or `SuperLearner` for the outcome regression models. If `method="mean"` this will be a list of length `length(ftypeOfInterest)` each of length `t0` (one regression for each failure type and for each timepoint). If `method="hazard"` this will be a list of length `length(ftypeOfInterest)` with one fit corresponding to the hazard for each cause of failure. If `returnModels = FALSE`, this entry will be `NULL`.

ctimeMod If `returnModels=TRUE` the fit object for the call to `glm` or `SuperLearner` for the pooled hazard regression model for the censoring distribution. If `returnModels=FALSE`, this entry will be `NULL`.

trtMod If `returnModels = TRUE` the fit object for the call to `glm` or `SuperLearner` for the conditional probability of `trt` regression model. If `returnModels = FALSE`, this entry will be `NULL`.

t0 The timepoint at which the function was evaluated.

ftime The numeric vector of failure times used in the fit.

ftype The numeric vector of failure types used in the fit.

trt The numeric vector of treatment assignments used in the fit.

adjustVars The data.frame of failure times used in the fit.

Examples

```
# simulate data
set.seed(1234)
n <- 200
trt <- rbinom(n, 1, 0.5)
adjustVars <- data.frame(W1 = round(runif(n)), W2 = round(runif(n, 0, 2)))

ftime <- round(1 + runif(n, 1, 4) - trt + adjustVars$W1 + adjustVars$W2)
ftype <- round(runif(n, 0, 1))

# Fit 1
# fit a survtmle object with glm estimators for treatment, censoring, and
# failure using the "mean" method
fit1 <- survtmle(ftime = ftime, ftype = ftype,
                 trt = trt, adjustVars = adjustVars,
                 glm.trt = "W1 + W2",
                 glm.ftime = "trt + W1 + W2",
                 glm.ctime = "trt + W1 + W2",
                 method = "mean", t0 = 6)
fit1

# Fit 2
# fit an survtmle object with SuperLearner estimators for failure and
# censoring and empirical estimators for treatment using the "mean" method
fit2 <- survtmle(ftime = ftime, ftype = ftype,
                 trt = trt, adjustVars = adjustVars,
                 SL.ftime = c("SL.mean"),
                 SL.ctime = c("SL.mean"),
                 method = "mean", t0 = 6)
fit2
```

`timepoints` *Evaluate Results over Time Points of Interest*

Description

Wrapper function for `survtmle` that takes a fitted `survtmle` object and computes the TMLE estimated incidence for all times specified in the `times` argument. For this function to work, the original call to `survtmle` should have been executed with `returnModels = TRUE`. This allows the function to be more efficient than repeated calls to `survtmle` in that `timepoints` will use fitted censoring (and hazard if `method="hazard"` was used in the original call) estimates. It is therefore advisable that the vector `times` used in the call to `timepoints` not include times beyond the time specified in `t0` in the original call to `survtmle`. This can be ensured be making the original call to `survtmle` with `t0 = max(ftime)`.

Usage

```
timepoints(object, times, returnModels = FALSE)
```

Arguments

object	A fitted survtmle object with returnModels = TRUE
times	The times to evaluate incidence.
returnModels	Should the function return fitted GLM or Super Learner models at each timepoint. If set to TRUE, memory issues could arise depending on the number of timepoints specified and the size of the Super Learner library.

Value

An object of class tp.survtmle with number of entries equal to length(times). Each entry is named "tX", where X denotes a single value of times.

Examples

```
# simulate data
set.seed(1234)
n <- 100
ftime <- round(runif(n, 1, 4))
ftype <- round(runif(n, 0, 2))
trt <- rbinom(n, 1, 0.5)
adjustVars <- data.frame(W1 = rnorm(n), W2 = rnorm(n))

# fit an initial survtmle object with t0=max(ftime)
fm <- survtmle(ftime = ftime, ftype = ftype,
               trt = trt, adjustVars = adjustVars,
               glm.trt = "1", glm.ftime = "trt + W1 + W2",
               glm.ctime = "trt + W1 + W2", method="mean",
               returnModels = TRUE)

# call timepoints to get cumulative incidence estimates at each timepoint
allTimes <- timepoints(object = fm, times = 1:4, returnModels = FALSE)

# look at results for time 1
class(allTimes$t1)
allTimes$t1
# look at results for time 2
allTimes$t2
```

---

updateVariables *Update TMLEs for Hazard to Cumulative Incidence*

Description

A helper function that maps hazard estimates into estimates of cumulative incidence and updates the "clever covariates" used by the targeted minimum loss-based estimation fluctuation step.

Usage

```
updateVariables(dataList, allJ, ofInterestJ, nJ, uniqtrt, ntrt, t0,
  verbose, ...)
```

Arguments

dataList	A list of data.frame objects.
allJ	Numeric vector indicating the labels of all causes of failure.
ofInterestJ	Numeric vector indicating ftypeOfInterest that was passed to hazard_tmle.
nJ	The number of unique failure types.
uniqtrt	The values of trtOfInterest passed to mean_tmle.
ntrt	The number of trt values of interest.
t0	The timepoint at which survtmle was called to evaluate.
verbose	A boolean indicating whether the function should print messages to indicate progress.
...	Other arguments. Not currently used.

Value

The function returns a list that is exactly the same as the input dataList, but with updated columns corresponding with estimated cumulative incidence at each time and estimated "clever covariates" at each time.

Index