Package ‘simMSM’

October 14, 2022

Type Package

Title Simulation of Event Histories for Multi-State Models

Version 1.1.42

Date 2022-05-06

Author Holger Reulen

Maintainer Holger Reulen <hreulen@uni-goettingen.de>

Description Simulation of event histories with possibly non-linear baseline hazard rate functions, non-linear (time-varying) covariate effect functions, and dependencies on the past of the history. Random generation of event histories is performed using inversion sampling on the cumulative all-cause hazard rate functions.

Depends survival, mvna

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-05-06 06:50:06 UTC

R topics documented:

mplskeleton *Build Up a Model Parameter List Skeleton*

Description

Constructs the skeleton of a model parameter list on basis of the transition-type definition matrix.

Usage

```
mplskeleton(tmat)
```

Arguments

`tmat`	a transition-type definition matrix. This is a square matrix containing the boolean information of which exit state-types (the columns) are reachable from which entry state-type (the lines).

Details

The example below provides an intuitive description of how to suitably set up the input argument.

Value

An incomplete (therefore the function name ends with 'skeleton') model parameter list as used for the input argument `mpl` in the function `simeventhistories`.

Author(s)

Holger Reulen

Examples

```
## Two state-type model with transient state-types 1 and 2:
tra2 <- matrix(ncol = 2, nrow = 2, data = FALSE)
tra2[1, 2] <- tra2[2, 1] <- TRUE
mplskeleton(tmat = tra2)
## Illness-death model (IDM) with recovery:
traIDM <- matrix(nrow = 3, ncol = 3, FALSE)
traIDM[1, 2] <- traIDM[1, 3] <- traIDM[2, 1] <- traIDM[2, 3] <- TRUE
mplskeleton(tmat = traIDM)
```

`plotbe` *Breslow Estimator of the Cumulative Baseline Hazard Rate Function*

Description

Calculates the Breslow estimator of the cumulative baseline hazard rate functions.

Usage

```
plotbe(m, mpl, return.be = FALSE, ...)
```

Arguments

`m`	estimated stratified `coxph` model.
`mpl`	model parameter list.
`return.be`	should a list containing the Breslow estimator values be returned?
`...`	further arguments and graphical parameters passed to `plot`, e.g. `xlim` for a re-specification of the shown time axis.

Details

The function is a specific wrapper function to the function `basehaz` from the R package `survival`.

Value

Plot of the Breslow estimator for the transition-type specific cumulative baseline hazard rate functions.

Author(s)

Holger Reulen

References

Therneau T (2014) A Package for Survival Analysis in S. R package version 2.37-7, http://CRAN.R-project.org/package=survival.

Terry M. Therneau and Patricia M. Grambsch (2000) Modeling Survival Data: Extending the Cox Model. Springer, New York. ISBN 0-387-98784-3.

Examples

```
## Not run: plotbe(d, mpl, return.be = FALSE, ...)
```

plotcph — *Cox PH Model Effect Estimates Illustration*

Description

Plot effects of a Cox proportional hazards model.

Usage

```
plotcph(m, ...)
```

Arguments

`m`	estimated stratified `coxph` model.
`...`	further arguments and graphical parameters passed to `plot`, as for example `ylim` for a re-specification of the shown covariate effect axis.

Details

The Cox proportional hazards model coefficients are illustrated by the solid black lines representing the estimated effect values (y axis) for the respective covariates (x axis), the grey polygons denote 95% confidence intervals.

Value

A plot.

Author(s)

Holger Reulen

References

Therneau T (2014) A Package for Survival Analysis in S. R package version 2.37-7, http://CRAN.R-project.org/package=survival.

Terry M. Therneau and Patricia M. Grambsch (2000) Modeling Survival Data: Extending the Cox Model. Springer, New York. ISBN 0-387-98784-3.

Examples

```
## Not run: plotcph(m, ...)
```

`plotnae` *Nelson-Aalen Estimators for Simulated Event History Data*

Description

Calculates the Nelson-Aalen estimators for the cumulative hazard rate functions for simulated event history data

Usage

```
plotnae(d, mpl, return.nae = FALSE, ...)
```

Arguments

`d`	simulated data-set list as the return object from the `simeventhistories` function.
`mpl`	model parameter list as provided to `simeventhistories`.
`return.nae`	should a list containing the values of the calculated Nelson-Aalen estimator be returned?
`...`	further arguments and graphical parameters passed to `plot.mvna`, e.g. `xlim` for a re-specification of the shown time axis, or `conf.int` for the plotting of pointwise confidence intervals.

Details

The function is a specific (w.r.t. to the structure of the result from `simeventhistories`) wrapper function to the function `mvna` from the same-named R package `mvna`.

Value

Plot of the Nelson-Aalen estimator and the underyling `mvna` result if `return.nae` is set to `TRUE`.

Author(s)

Holger Reulen

References

Allignol, A., Beyersmann, J., Schumacher, M. (2008) *mvna: An R Package for the Nelson-Aalen Estimator in Multistate Models*, R News, 8 (2): 48 – 50

See Also

`mplskeleton`, `simeventhistories`

Examples

```
## Not run: plotnae(d, mpl, return.nae = FALSE, ...)
```

pmeskeleton *Build Up a Partial Markov Model Linear Predictor List Skeleton*

Description

Constructs the skeleton of a linear predictor list for partial Markov influences on basis of the transition-type definition matrix.

Usage

```
pmeskeleton(tmat)
```

Arguments

`tmat`	a transition-type definition matrix. This is a square matrix containing the boolean information of which exit state-types (the columns) are reachable from which entry state-type (the lines).

Details

The example below provides an intuitive description of how to suitably set up the input arguments.

Value

An incomplete (therefore the function name ends with 'skeleton') linear predictor list as used for the `partial.markov.eta` input argument in the function `simeventhistories`.

Author(s)

Holger Reulen

Examples

```
## Two state-type model with transient state-types 1 and 2:
tra2 <- matrix(ncol = 2, nrow = 2, data = FALSE)
tra2[1, 2] <- tra2[2, 1] <- TRUE
pmeskeleton(tmat = tra2)
## Illness-death model (IDM) with recovery:
traIDM <- matrix(nrow = 3, ncol = 3, FALSE)
traIDM[1, 2] <- traIDM[1, 3] <- traIDM[2, 1] <- traIDM[2, 3] <- TRUE
pmeskeleton(tmat = traIDM)
```

`simeventhistories` *Simulate Event Histories*

Description

Simulates n individual event histories.

Usage

```
simeventhistories(n, mpl, max.time, change.times, X, states.at.origin = NULL,
Xstruc, partial.markov.x = NULL, partial.markov.eta = NULL)
```

Arguments

`n`	number of individuals.
`mpl`	model parameter list as generated (only a skeleton that has to be suitably completed) by the function `mplskeleton` (see examples below).
`max.time`	maximum entry time.
`change.times`	vector giving the times of change of the time-change covariates.
`X`	design matrix.
`states.at.origin`	state-types at origin (default is all possible entry state-types, which is internally calculated).
`Xstruc`	X structure matrix. See Examples for more information.
`partial.markov.x`	function defining how the partial Markov covariates are generated (see example below).
`partial.markov.eta`	list of lists (as generated by the function `pmeskeleton` in close analogy to `mpl`) defining how the partial Markov linear predictors are generated (see example below).

Details

The example below provides an intuitive description of how to use the different input arguments. The idea of partial Markov covariates is based on the definition in Commenges (1991). A description of this idea directly in the context of illness-death models is described on pp. 224-225 in Beyersmann et al. (1999).

Value

Three data frames named `msm.bascis`, `ttsce`, `tt.indicators` are returned organized within one list. The three data frames and their respective variables will be described in the next lines.

`msm.bascis` contains the following variables variables:

`id`	id (1, ..., n) of the individual

`entry`	entry times
`exit`	exit times
`from`	values of initial states
`to`	values of final states
`delta`	non-censoring indicator function
`x1`	values of first covariate (additional covariates follow). If partial Markov objects are supplied, the generated covariates are attached as additional variables.

The second data frame `ttsce` contains a transition-type specific covariate expansion (as well for partial Markov covariates in the case of a partial Markov set-up).

The third data frame `tt.indicators` contains the values of transition-type indicator functions. For censored observations, all values of one data line are equal to zero (as e.g. needed in a BayesX full likelihood analysis).

Author(s)

Holger Reulen

References

Daniel Commenges (1991) Multi-state Models in Epidemiology. Lifetime Data Analysis, Vol. 5, No. 4.

Jan Beyersmann, Martin Schumacher, Arthur Allignol (2012) Competing Risks and Multistate Models with R. Springer Series 'UseR!'.

See Also

`mplskeleton`

Examples

```
## An example for a time-varying setup without partial Markov effects:
tra2 <- matrix(ncol = 2, nrow = 2, data = FALSE)
tra2[1, 2] <- tra2[2, 1] <- TRUE
mpl <- mplskeleton(tmat = tra2)
mpl[[1]]$bhr[[2]] <- mpl[[2]]$bhr[[1]] <- function(t){return(0.5)}
mpl[[1]]$eta[[2]] <- function(x.i, t){ ## time-varying x2 and time-varying f(x2)
  ifelse(t < 5,
         return(1.0 * x.i[1] + 0.5 * x.i[2]),
         return(1.0 * x.i[1] + 1.0 * x.i[3]))}
mpl[[2]]$eta[[1]] <- function(x.i, t){ ## time-varying x2 and time-varying f(x1)
  ifelse(t < 5,
         return(-0.5 * x.i[1] + 0.5 * x.i[2]),
         return( 1.0 * x.i[1] + 0.5 * x.i[3]))}
set.seed(123)
N <- 2
X <- matrix(nrow = N, ncol = 2, rnorm(2 * N))
X <- cbind(X, X[, 2] + runif(N)/10)
colnames(X) <- c("x1", "x2.t1", "x2.t2")
```

```
Xstruc <- matrix(ncol = 2, nrow = 2, data = 0)
rownames(Xstruc) <- c("t1", "t2")
colnames(Xstruc) <- c("x1", "x2")
Xstruc[, 1] <- 1
Xstruc[, 2] <- c(2, 3)
d <- simeventhistories(n = N, mpl = mpl, X = X, max.time = 10,
                       change.times = c(5), Xstruc = Xstruc)
head(d$msm.basics)
## Not run:
## An Illness-Death model example with time-varying setup and partial Markov
## effects:
traIDM <- matrix(nrow = 3, ncol = 3, FALSE)
traIDM[1, 2] <- traIDM[1, 3] <- traIDM[2, 1] <- traIDM[2, 3] <- TRUE
mpl <- mplskeleton(tmat = traIDM)
mpl[[1]]$bhr[[2]] <- mpl[[1]]$bhr[[3]] <- mpl[[2]]$bhr[[1]] <-
                     mpl[[2]]$bhr[[3]] <- function(t){0.25}
mpl[[1]]$eta[[2]] <- mpl[[1]]$eta[[3]] <- mpl[[2]]$eta[[1]] <-
  mpl[[2]]$eta[[3]] <- function(x.i, t){
    ifelse(t < 5,
      return(0.5 * x.i[1]),
      return(0.5 * x.i[2]))}
set.seed(123)
N <- 500
X <- matrix(nrow = N, ncol = 1, rnorm(N))
X <- cbind(X, X[, 1] + rnorm(N)/10)
colnames(X) <- c("x1.t1", "x1.t2")
Xstruc <- matrix(ncol = 1, nrow = 2, data = 0)
rownames(Xstruc) <- c("t1", "t2")
colnames(Xstruc) <- c("x1")
Xstruc[, 1] <- c(1, 2)
Xstruc
## Now set-up the partial Markov influences:
## Function 'partial.markov.x' has to take 5 input arguments representig vectors
## of past history information. They have to take names 'entry', 'exit', 'from',
## 'to', and 'delta':
partial.markov.x <- function(entry, exit, from, to, delta){
  count.12 <- sum(as.numeric((from == 1) & (to == 2) & (delta == 1)))
  count.21 <- sum(as.numeric((from == 2) & (to == 1) & (delta == 1)))
  return(c(count.12, count.21))}
## List 'partial.markov.eta' is a list of lists in analogy to 'mpl':
partial.markov.eta <- pmeskeleton(traIDM)
partial.markov.eta[[1]][[2]] <- function(x){return( 0.25 * x[1])}
partial.markov.eta[[1]][[3]] <- function(x){return( 0.50 * x[1])}
partial.markov.eta[[2]][[1]] <- function(x){return(-0.50 * x[1] + 0.25 * x[2])}
partial.markov.eta[[2]][[3]] <- function(x){return(0)}
## Event history simulation:
d <- simeventhistories(n = N, mpl = mpl, X = X, max.time = 10,
                       change.times = c(5), Xstruc = Xstruc,
                       partial.markov.x = partial.markov.x,
                       partial.markov.eta = partial.markov.eta)

## End(Not run)
```

tolongformat *Transforms Data Frame into Long Format Design*

Description

Data frame with one line per event gets transformed to a data frame in a format that has as many rows as each subject has transitions for which he/she is at risk.

Usage

```
tolongformat(d, mpl)
```

Arguments

`d`	simulated data-set as the return object from the `simeventhistories` function.
`mpl`	model parameter list.

Details

In the format of the input data frame object `d`, the data are not yet suitable for a stratified Cox partial likelihood analysis: we need the data frame in a format that has many rows as each subject has transitions for which he/she is at risk. We will denote this as 'long format' in reference to the literature on multi-state model software, as for example on page 5 in de Wreede et al (2011).

Value

A list of data-sets.

Author(s)

Holger Reulen

References

Liesbeth C. de Wreede, Marta Fiocco, Hein Putter (2011) mstate: An R Package for the Analysis of Competing Risks and Multi-State Models. Journal of Statistical Software, 38(7), 1-30. URL http://www.jstatsoft.org/v38/i07/.

See Also

`simeventhistories`

Examples

```
## Not run: tolongformat(d, mpl)
```

Index