Package 'gems'

October 13, 2022

Type Package

Title Generalized Multistate Simulation Model

Version 1.1.1

Date 2017-03-26

Author Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

Maintainer Luisa Salazar Vizcaya <luisapaola.salazarvizcaya@insel.ch>

Imports MASS, methods, msm, plyr, graphics, stats, utils, data.table

Suggests muhaz

Description Simulate and analyze multistate models with general hazard
functions. gems provides functionality for the preparation of hazard functions
and parameters, simulation from a general multistate model and predicting future
events. The multistate model is not required to be a Markov model and may take
the history of previous events into account. In the basic version, it allows
to simulate from transition-specific hazard function, whose parameters are
multivariable normally distributed.

License GPL-2

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-03-26 16:02:51 UTC

R topics documented:

ArtCohort   *Class* "ArtCohort"

Description

Is a S4 class for the artificial cohort generated by `simulateCohort`.

Usage

```
## S4 method for signature 'ArtCohort,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'ArtCohort'
update(object, newsize, addbaseline = matrix(NA, nrow =
  newsize - object@size), newInitialStates = rep(1, newsize - object@size))

## S4 method for signature 'ArtCohort'
head(x, ...)

## S4 method for signature 'ArtCohort'
tail(x, ...)

## S4 method for signature 'ArtCohort'
summary(object)
```

Arguments

`x, object`	an ArtCohort
`i, j, drop`	same as for `data.frame`
`...`	passed on to `data.frame` method
`newsize`	size of the updated cohort
`addbaseline`	baseline for new part of cohort
`newInitialStates`	
	initial states for new part of cohort

Slots

`states.number` Object of class "numeric": number of states

`size` Object of class "numeric": cohort size

`baseline` Object of class "matrix": baseline matrix

follow.up Object of class "numeric": maximum follow-up time

parameters Object of class "transition.structure": input parameters

parametersCovariances Object of class "transition.structure": input covariance matrices

timeToTransition Object of class "matrix": input timeToTransition matrix. logical components

transitionFunctions Object of class "transition.structure": input hazard functions

time.to.state Object of class "data.frame": entry times for each patient into each of the states

Objects from the Class

Objects are created by calls to the function simulateCohort.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

See Also

[simulateCohort](), [transition.structure](), [transitionProbabilities](), [cumulativeIncidence]()

Examples

```
showClass("ArtCohort")
```

---

cumulativeIncidence *transition probabilities*

Description

Calculates the cumulative incidence and prediction intervals after first state

Usage

```
cumulativeIncidence(object, times, M = 100, stateNames = paste("State",
  as.list(1:dim(cohorts)[1])))
```

Arguments

object	either the output of [simulateCohort]() or the matrix with the probabilities slot of that output.
times	a time vector.
M	number of groups for calculating confidence intervals.
stateNames	a list with the names of states.

Value

an object of class "PosteriorProbabilities", containing the statenames, timepoints and the cumulative incidence with pointwise prediction intervals over time.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

See Also

PosteriorProbabilities, ArtCohort, simulateCohort

---

gems   *gems: Generalized Multistate Simulation Model*

Description

Simulate and analyze multistate models with general hazard functions. gems provides functionality for the preparation of hazard functions and parameters, simulation from a general multistate model and predicting future events. The multistate model is not required to be a Markov model and may take the history of previous events into account. In the basic version, it allows to simulate from transition-specific hazard function, whose parameters are multivariable normally distributed.

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

---

generateHazardMatrix   *generate template for transition functions*

Description

This function simplifies generating the matrix of transition functions.

Usage

```
generateHazardMatrix(statesNumber)
```

Arguments

statesNumber   the number of states to be considered.

Value

a `transition.structure` of dimension $N \times N$, where N is the number of states and with value "impossible" for all potential transitions.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

See Also

[transition.structure](), [simulateCohort]()

---

generateParameterCovarianceMatrix

generate a template for parameter covariances

Description

This function simplifies generating the matrix of parameter covariances from a matrix of mean parameters.

Usage

```
generateParameterCovarianceMatrix(mu)
```

Arguments

mu   a `transition.structure` of dimension $N \times N$, whose components `list` the mean values for the parameters in the `transitionFunction`.

Value

a `transition.structure` of dimension $N \times N$ of covariance matrices for the `parameters`.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

See Also

transition.structure, generateParameterMatrix, simulateCohort

---

generateParameterMatrix

generate a template for mean parameters

Description

This function simplifies generating the matrix of mean parameters from a matrix of transition functions.

Usage

```
generateParameterMatrix(hf)
```

Arguments

hf   a transition.structure of dimension $N \times N$, where N is the number of states.

Value

a transition.structure of dimension $N \times N$, whose components are lists of the right length for the parameters in the corresponding hazard function hf.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

See Also

transition.structure, simulateCohort

PosteriorProbabilities

Class "PosteriorProbabilities"

Description

This S4 class summarizes the posterior probabilities over time for objects of class "ArtCohort"

Usage

```
## S4 method for signature 'PosteriorProbabilities,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'PosteriorProbabilities'
plot(x, ci = FALSE, main = paste(x@type,
  "after starting in State", x@states[1], "at time 0"),
  states = 1:dim(x@probabilities)[2], lwd = c(2, 2), col = c("blue",
  "green3"), lty = c(1, 2), xlab = "Time", ylab = "Probability", ...)

## S4 method for signature 'PosteriorProbabilities'
head(x, ...)

## S4 method for signature 'PosteriorProbabilities'
tail(x, ...)
```

Arguments

x	the PosteriorProbabilities object
i, j, drop	same as for a "data.frame"
...	arguments passed on to main method
ci	should confidence intervals be displayed
main, xlab, ylab	
	same as any plot
states	which states to display
lwd, col, lty	vectors of lentht 2, with first component for the point estimate and second component for the confidence interval

Slots

states Object of class "character": names of states

times Object of class "numeric": time points at which probabilities are evaluated

probabilities Object of class "matrix": posterior Probabilities to be in each state at each time

lower Object of class "matrix": lower prediction bound to be in each state at each time

upper Object of class "matrix": upper prediction bound to be in each state at each time

type Object of class "character": describes type of probability

Objects from the Class

Objects are created by calls to the function `simulateCohort`.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

See Also

`transitionProbabilities`, `cumulativeIncidence`, `ArtCohort`

Examples

```
showClass("PosteriorProbabilities")
```

---

simulateCohort   *Simulate cohort*

Description

Simulates a cohort of patients from a set of functions associated to each possible transition in a multistate model. The multistate model is not required to be a Markov model and may take the history of previous events into account. In the basic version, it allows to simulate from transition-specific hazard function, whose parameters are multivariable normally distributed. For each state, all transition-specific hazard functions and their parameters need to be specified. For simulating one transition, all possible event times are simulated and the minimum is chosen. Then simulation continues from the corresponding state until an absorbing state of time `to` is reached.

Usage

```
simulateCohort(transitionFunctions, parameters, cohortSize = 1000,
  parameterCovariances = FALSE, timeToTransition = array(FALSE, dim =
  dim(transitionFunctions@list.matrix)), baseline = matrix(NA, nrow =
  cohortSize), initialState = rep(1, cohortSize),
  absorbing = transitionFunctions@states.number, to = 100,
  report.every = 100, sampler.steps = 1000)
```

Arguments

transitionFunctions
                a `transition.structure` of dimension $N \times N$ that contains the hazard functions
                tions

parameters   a `transition.structure` of dimension $N \times N$ that contains the parameters

cohortSize   a `numeric` indicating the number of patients to be simulated.

parameterCovariances

> a `transition.structure` of dimension $N \times N$ of covariance matrices for the `parameters`.

timeToTransition

> a `logical matrix`; TRUE for all transitions whose `transitionFunction` is specified as the time until transition instead of as a hazard function or as a `character`.

baseline   a `matrix` or `data.frame` of dimension $cohortSize \times M$ with M baseline characteristics of subjects to be simulated.

initialState a `numeric` of length `cohortSize` with the initial state for each subject simulated.

absorbing   a `numeric` containing all absorbing states.

to   final time of the simulation.

report.every a `numeric` to check progress of simulation.

sampler.steps a `numeric` indicating number of steps for discretization of hazard functions

Details

The `transitionFunctions` contains hazard functions or time to event function associated to each possible transition. The elements of this `list` can be either expressed as an explicit R `function` or as a `character` ("impossible", "Weibull", "multWeibull", "exponential") in order to express impossible transitions or parametric forms for the distributions of time to event. If the functions should depend on time, baseline characteristics or be *history-dependent*, the function arguments *t*, *bl* or *history* can be used. Time *t* refers to the time since entry into the current state. For the time since the initial state, use `t+sum(history)`.

The components of the `parameters` argument `list` the mean values for the parameters in the `transitionFunction`. If the corresponding `transitionFunction` is a `function`, the parameters should appear in the same order as in the `function`, leaving out *t*, *bl* and *history*. If the corresponding `transitionFunction` is the `character` "Weibull", the first argument is the shape and the second one the scale. If the corresponding `transitionFunction` is the `character` "multWeibull", specify weights, shapes, scales in this order.

Note that when using the `parameterCovariances` argument it is the users responsibility to ensure that the functions are parametrized such that `parameters` for each transition are multivariate normally distributed and mutually independent.

Value

an object of class `"ArtCohort"` with `time.to.state` slot of dimension $cohortSize \times N$ with entry times for each patient into each of the states.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

simulateCohort

See Also

generateHazardMatrix, generateParameterMatrix, generateParameterCovarianceMatrix,
ArtCohort, transitionProbabilities, cumulativeIncidence

Examples

```
# Here is an example model with 3 states and 2 possible transitions.

# number of states in the model
statesNumber <- 3

# cohort size
cohortSize <- 100

# specification of hazard functions
hazardf <- generateHazardMatrix(statesNumber)
hazardf[[1,2]] <- function(t, r1, r2)
{
  ifelse(t<=2, r1 , r2)
}
hazardf[[2,3]] <- "Weibull"

# list of parameters for the hazard functions
mu <- generateParameterMatrix(hazardf)
mu[[1,2]] <- list(0.33,  0.03) # r1, r2
mu[[2,3]] <- list(1,0.84) # shape, scale

# time
maxTime <- 10

# simulate the cohort
cohort <- simulateCohort(
  transitionFunctions = hazardf,
  parameters = mu,
  cohortSize = cohortSize,
  to=maxTime)

# output
head(cohort)

# transition probability
tr <- transitionProbabilities(cohort, times=seq(0,4,.1))
plot(tr, ci=FALSE)

# cumulative incidence
inc <- cumulativeIncidence(cohort, times=seq(0,4,.1))
plot(inc, ci=FALSE, states=c(2,3))
```

---

tavi *tavi data set*

Description

The simulated data set for each patient contains data for kidney injuries, bleeding complications and the combined endpoint of stroke or death. The data was simulated from the original data following the steps described in the package vignette.

Format

A data frame with 194 observations on the following 7 variables.

id a character vector that contains the patient id's

kidney a numeric vector; indicator variable that show if an event has occurred

kidney.dur a numeric vector; times at which the events occurred or the patients were censored

bleeding a numeric vector; indicator variable that show if an event has occurred

bleeding.dur a numeric vector; times at which the events occurred or the patients were censored

death a numeric vector; indicator variable that show if an event has occurred

death.dur a numeric vector; times at which the events occurred or the patients were censored

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

Examples

```
head(data(tavi))
```

---

transition.structure *Class* "transition.structure"

Description

This S4 class provides a structure to specify different characteristics of transitions, such as transition functions functions, parameters or parameter covariances.

Usage

```
## S4 method for signature 'transition.structure'
x[[i, j, ..., exact = TRUE]]

## S4 replacement method for signature 'transition.structure'
x[[i, j]] <- value

possibleTransitions(object)

## S4 method for signature 'transition.structure'
possibleTransitions(object)

## S4 method for signature 'transition.structure'
print(x)
```

Arguments

x, object   the `transition.structure`

i, j   same as for `matrix`

exact, value, ...

                 passed on to `list` method

Slots

`states.number` Object of class `"numeric"`: number of states

`list.matrix` Object of class `"matrix"`: a list with two dimensions, where list element [i,j]
    correspond to transitions from i to j

Objects from the Class

Objects are created by calls to the functions generateHazardMatrix, generateParameterMatrix,
`generateParameterCovarianceMatrix`.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

See Also

[generateHazardMatrix](#), [generateParameterMatrix](#), [generateParameterCovarianceMatrix](#)

Examples

```
showClass("transition.structure")
```

---

transitionProbabilities

transition probabilities

Description

Calculates the probabilities and prediction intervals after first state

Usage

```
transitionProbabilities(object, times, M = 100, stateNames = paste("State",
    as.list(1:dim(cohorts)[1])))
```

Arguments

object	either the output of [simulateCohort](#) or the `matrix` with the `probabilities` slot of that output.
times	a time vector.
M	number of groups for calculating confidence intervals.
stateNames	a list with the names of states.

Value

an object of class `"PosteriorProbabilities"`, containing the statenames, timepoints and the transition probabilities with pointwise prediction intervals over time.

Author(s)

Luisa Salazar Vizcaya, Nello Blaser, Thomas Gsponer

References

Nello Blaser, Luisa Salazar Vizcaya, Janne Estill, Cindy Zahnd, Bindu Kalesan, Matthias Egger, Olivia Keiser, Thomas Gsponer (2015). gems: An R Package for Simulating from Disease Progression Models. Journal of Statistical Software, 64(10), 1-22. URL http://www.jstatsoft.org/v64/i10/.

See Also

[PosteriorProbabilities](#), [ArtCohort](#), [simulateCohort](#)

Index