

Package ‘multistateutils’

December 15, 2019

Type Package

Title Utility Functions for Parametric Multi-State Models

Version 1.2.4

Date 2019-12-14

Author Stuart Lacy

Maintainer Stuart Lacy <stuart.lacy@gmail.com>

Description Provides functions for working with multi-state modelling, such as efficient simulation routines for estimating transition probabilities and length of stay. It is designed as an extension to multi-state modelling capabilities provided with the 'flexsurv' package (see Jackson (2016) <doi:10.18637/jss.v070.i08>).

License GPL (>= 3)

Imports data.table, dplyr, magrittr, networkD3, tidyr

Suggests covr, knitr, flexsurv, mstate, microbenchmark, rmarkdown, testthat

URL <https://github.com/stulacy/multistateutils>

BugReports <https://github.com/stulacy/multistateutils/issues>

LinkingTo Rcpp

RoxygenNote 7.0.2

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-12-15 16:50:07 UTC

R topics documented:

cohort_simulation	2
length_of_stay	3
msprep2	5
multistateutils	7
plot_predicted_pathway	7
predict_transitions	8

cohort_simulation	<i>Runs a cohort discrete event simulation</i>
-------------------	------------------------------------------------

Description

Runs a cohort discrete event simulation

Usage

```
cohort_simulation(
  models,
  newdata,
  trans_mat,
  start_time = 0,
  start_state = 1,
  time_limit = NULL,
  tcovs = NULL,
  M = 1000,
  ci = FALSE,
  ci_margin = 0.95,
  agelimit = FALSE,
  agecol = "age",
  agescale = 365.25
)
```

Arguments

models	List of flexsurvreg objects.
newdata	Data frame with covariates of individual to simulate times for. Must contain all fields required by models.
trans_mat	Transition matrix, such as that used in mstate.
start_time	Entry times of individuals specified in newdata. Can either be a single time that everyone enters at, or have as many values as rows in newdata. Defaults to everyone starting at time 0.
start_state	The starting state of the individuals specified in newdata. States can be represented by an integer (the row/column number of the state in trans_mat), or as a string giving the name of the state in trans_mat. Can either be a single value when everyone starts in the same state, or have as many values as rows in newdata. Defaults to everyone starting in state 1.
time_limit	The maximum time to run the simulation for. If not provided then the simulation runs until all the individuals have obtained a sink state.
tcovs	As in flexsurv::pmatrix.simfs, this is the names of covariates that need to be incremented by the simulation clock at each transition, such as age when modelled as age at state entry.

M	Number of times to run the simulations in order to obtain confidence interval estimates.
ci	Whether to calculate confidence intervals. See <code>flexsurv::pmatrix.simfs</code> for details.
ci_margin	Confidence interval range to use if <code>ci</code> is set to TRUE.
agelimit	Whether to automatically assign people to an 'early death' state. This is useful as otherwise individuals can be assigned unrealistic time-to-events due to the nature of sampling times from a random number distribution. If this value is FALSE then no limit is applied, otherwise provide the time-limit to be used. This limit must be in the same time-scale as the time-to-event models.
agecol	The name of the column in <code>newdata</code> that holds an individual's age.
agescale	Any multiplication to be applied to the age covariate to put it onto the same time-scale as the simulation. This is often useful as time-to-event may be measured on a day-based time-scale while age is typically measured in years.

Value

A data frame with state entry times for each individual.

Examples

```
library(multistateutils)
library(mstate)
library(flexsurv)

# Convert data to long
data(ebmt3)
tmat <- trans.illdeath()
long <- msprep(time=c(NA, 'prtime', 'rfstime'),
               status=c(NA, 'prstat', 'rfsstat'),
               data=ebmt3,
               trans=tmat,
               keep=c('age', 'dissub'))

# Fit parametric models
models <- lapply(1:3, function(i) {
  flexsurvreg(Surv(time, status) ~ age + dissub, data=long, dist='weibull')
})

sim <- cohort_simulation(models, ebmt3, tmat)
```

length_of_stay	<i>Estimates length of stay</i>
----------------	---------------------------------

Description

Estimates length of stay in each state of an individual's passage through a multi-state model by discrete event simulation.

Usage

```
length_of_stay(
  models,
  newdata,
  trans_mat,
  times,
  start_state = 1,
  tcovs = NULL,
  N = 1e+05,
  M = 1000,
  ci = FALSE,
  ci_margin = 0.95,
  agelimit = FALSE,
  agecol = "age",
  agescale = 365.25
)
```

Arguments

models	List of flexsurvreg objects.
newdata	Data frame with covariates of individual to simulate times for. Must contain all fields required by models.
trans_mat	Transition matrix, such as that used in mstate.
times	Times at which to estimate length of stay.
start_state	Starting state. Either number or character name in trans_mat.
tcovs	As in flexsurv::pmatrix.simfs, this is the names of covariates that need to be incremented by the simulation clock at each transition, such as age when modelled as age at state entry.
N	Number of times to repeat the individual
M	Number of times to run the simulations in order to obtain confidence interval estimates.
ci	Whether to calculate confidence intervals. See flexsurv::pmatrix.simfs for details.
ci_margin	Confidence interval range to use if ci is set to TRUE.
agelimit	Whether to automatically assign people to an 'early death' state. This is useful as otherwise individuals can be assigned unrealistic time-to-events due to the nature of sampling times from a random number distribution. If this value is FALSE then no limit is applied, otherwise provide the time-limit to be used. This limit must be in the same time-scale as the time-to-event models.
agecol	The name of the column in newdata that holds an individual's age.
agescale	Any multiplication to be applied to the age covariate to put it onto the same time-scale as the simulation. This is often useful as time-to-event may be measured on a day-based time-scale while age is typically measured in years.

Value

A data frame containing length of stay estimates.

Examples

```
library(multistateutils)
library(mstate)
library(flexsurv)

# Convert data to long
data(ebmt3)
tmat <- trans.illdeath()
long <- msprep(time=c(NA, 'prtime', 'rfsstime'),
               status=c(NA, 'prstat', 'rfsstat'),
               data=ebmt3,
               trans=tmat,
               keep=c('age', 'dissub'))

# Fit parametric models
models <- lapply(1:3, function(i) {
  flexsurvreg(Surv(time, status) ~ age + dissub, data=long, dist='weibull')
})

# New individual to estimate transition probabilities for
newdata <- data.frame(age="20-40", dissub="AML")

# Estimate length of stay in each state after a year, given starting in state 1
length_of_stay(models,
               newdata=newdata,
               tmat, times=365.25,
               start=1)
```

msprep2

Converts long state entry data into a format suitable for multi-state modelling

Description

This function performs the same role as `msprep` from the `mstate` package, except that it accepts long data (each row corresponds to a state entry) rather than the wide format used by `msprep` (each row represents an individual with state entry indicated in columns).

Usage

```
msprep2(
  entry,
  tmat,
```

```

censors = NULL,
start_times = NULL,
start_states = NULL,
covars = NULL,
idcol = "id"
)

```

Arguments

entry	Long data frame of format <code>id state time</code> . State can either be character, with the same state names used in <code>tmat</code> , or integer where they refer to the rownumber of that state in <code>tmat</code> . Note that the <code>state</code> and <code>time</code> fields must be labelled this way.
tmat	Transition matrix in the standard format required by <code>msprep</code> .
censors	A long data frame with 2 columns, <code>id</code> and <code>sensor_time</code> . Gives the last follow-up time for individuals that haven't entered a sink state.
start_times	A long data frame with 2 columns, <code>id</code> and <code>start_time</code> . Gives the time at which the patient entered the simulation. Defaults to 0.
start_states	A long data frame with 2 columns, <code>id</code> and <code>start_state</code> . Gives the state the patient entered the simulation in. Defaults to 1.
covars	Data frame where each row corresponds to an individual and details their covariate values. Must contain the <code>id</code> column specified in <code>idcol</code> alongside any covariate fields of interest.
idcol	The column that indexes these patients, must be present in <code>entry</code> and <code>censors</code> , <code>start_times</code> , <code>start_states</code> , and <code>covars</code> if supplied.

Details

The long data format required by `msprep2` is a more natural way of organising state entry data than the wide format required by `msprep`. An additional benefit of having the state entries organised in this fashion is that it allows for the situation where an individual enters the same state multiple times, which is not supported by `msprep`.

Value

An object of class `msdata`, representing the transition data in long format.

Examples

```

library(multistateutils)
library(mstate)

tmat <- trans.illdeath()
entry <- data.frame(id=c(1, 2, 2),
                    state=c(2, 2, 3),
                    time=c(23, 35, 360))

msprep2(entry, tmat)

```

multistateutils	multistateutils <i>package</i> .
-----------------	----------------------------------

Description

Provides utility functions for already fitted parametric multi-state models, such as estimating transition probabilities and length of stay, as well as means of visualising these predictions.

plot_predicted_pathway	<i>Displays the predicted flow for a given individual through the system</i>
------------------------	------------------------------------------------------------------------------

Description

Produces an interactive HTML widget that displays a Sankey diagram showing a predicted patient pathway through the multi-state model.

Usage

```
plot_predicted_pathway(
  models,
  trans_mat,
  newdata,
  times,
  starting_state = 1,
  tcovs = NULL
)
```

Arguments

models	List of flexsurvreg objects.
trans_mat	Transition matrix, such as that used in <code>mstate</code> .
newdata	A data frame containing the attributes of the person to display the predicted state flow for. As the diagram can only be displayed for a single individual it will ignore any rows after the first.
times	The time-points at which to estimate transition probabilities.
starting_state	Starting state. Either number or character name in <code>trans_mat</code> .
tcovs	As in <code>flexsurv::pmatrix.simfs</code> , this is the names of covariates that need to be incremented by the simulation clock at each transition, such as age when modelled as age at state entry.

Value

The HTML widget.

Examples

```
library(multistateutils)
library(mstate)
library(flexsurv)

# Convert data to long
data(ebmt3)
tmat <- trans.illdeath()
long <- msprep(time=c(NA, 'prtime', 'rfstime'),
               status=c(NA, 'prstat', 'rfsstat'),
               data=ebmt3,
               trans=tmat,
               keep=c('age', 'dissub'))

# Fit parametric models
models <- lapply(1:3, function(i) {
  flexsurvreg(Surv(time, status) ~ age + dissub, data=long, dist='weibull')
})

# New individual to estimate transition probabilities for
newdata <- data.frame(age="20-40", dissub="AML")

# Plot pathway diagram at 2-yearly intervals up to 10-years
time_points <- seq(0, 10, by=2) * 365.25

plot_predicted_pathway(models, tmat, newdata, time_points, 1)
```

predict_transitions *Estimates transition probabilities*

Description

Estimates transition probabilities of an individual's passage through a multi-state model by discrete event simulation.

Usage

```
predict_transitions(
  models,
  newdata,
  trans_mat,
  times,
  start_times = 0,
  tcovs = NULL,
  N = 1e+05,
```



```

M = 1000,
ci = FALSE,
ci_margin = 0.95,
agelimit = FALSE,
agecol = "age",
agescale = 365.25
)

```

Arguments

models	List of flexsurvreg objects.
newdata	Data frame with covariates of individual to simulate times for. Must contain all fields required by models.
trans_mat	Transition matrix, such as that used in mstate.
times	Times at which to estimate transition probabilities.
start_times	Conditional time for transition probability.
tcovs	As in flexsurv::pmatrix.simfs, this is the names of covariates that need to be incremented by the simulation clock at each transition, such as age when modelled as age at state entry.
N	Number of times to repeat the individual
M	Number of times to run the simulations in order to obtain confidence interval estimates.
ci	Whether to calculate confidence intervals. See flexsurv::pmatrix.simfs for details.
ci_margin	Confidence interval range to use if ci is set to TRUE.
agelimit	Whether to automatically assign people to an 'early death' state. This is useful as otherwise individuals can be assigned unrealistic time-to-events due to the nature of sampling times from a random number distribution. If this value is FALSE then no limit is applied, otherwise provide the time-limit to be used. This limit must be in the same time-scale as the time-to-event models.
agecol	The name of the column in newdata that holds an individual's age.
agescale	Any multiplication to be applied to the age covariate to put it onto the same time-scale as the simulation. This is often useful as time-to-event may be measured on a day-based time-scale while age is typically measured in years.

Value

A data frame with estimates of transition probabilities.

Examples

```

library(multistateutils)
library(mstate)
library(flexsurv)

```

```
# Convert data to long
data(ebmt3)
tmat <- trans.illdeath()
long <- msprep(time=c(NA, 'prtime', 'rfstime'),
              status=c(NA, 'prstat', 'rfsstat'),
              data=ebmt3,
              trans=tmat,
              keep=c('age', 'dissub'))

# Fit parametric models
models <- lapply(1:3, function(i) {
  flexsurvreg(Surv(time, status) ~ age + dissub, data=long, dist='weibull')
})

# New individual to estimate transition probabilities for
newdata <- data.frame(age="20-40", dissub="AML")

# Estimate transition probabilities at 1 year
predict_transitions(models, newdata, tmat, times=365)

# Estimate transition probabilities at 1 year given we know they're alive after 6 months
predict_transitions(models, newdata, tmat, times=365, start_times = 365/2)
```

Index

cohort_simulation, 2

length_of_stay, 3

msprep2, 5

multistateutils, 7

plot_predicted_pathway, 7

predict_transitions, 8