

Package ‘spm’

December 31, 2015

Type Package

Title Stochastic Process Model (SPM)

Version 1.0

Date 2015-10-05

Author I. Y. Zhbannikov, I. V. Akushevich, K. G. Arbeev, A. I. Yashin

Maintainer Ilya Y. Zhbannikov <i.zhbannikov@mail.ru>

Description Stochastic Process Modeling

License GPL

Imports Rcpp (>= 0.11.1),
RcppArmadillo (>= 0.4.200.0)

LinkingTo Rcpp, RcppArmadillo

Depends deSolve,mice,sas7bdat,RcppArmadillo

Suggests knitr

VignetteBuilder knitr

RoxygenNote 5.0.1

R topics documented:

prepare_data	2
simdata_cont	3
simdata_cont_1D	4
simdata_discr	5
simdata_time_dep	6
spm	6
spm_continuous	7
spm_continuous_1D	8
spm_discrete	9
spm_time_dep	10
Index	12

prepare_data	<i>Data pre-processing for analysis with stochastic process model methodology.</i>
--------------	--

Description

Data pre-processing for analysis with stochastic process model methodology.

Usage

```
prepare_data(x, y, col.id = NULL, col.status = NULL, col.age = NULL,
  col.age.event = NULL, covariates = NULL, interval = 1,
  verbose = FALSE)
```

Arguments

x	A path to the table with follow-up oservations (longitudinal study). Formats: csv, sas7bdat
y	A path to the table with vital statistics (mortality). File formats: csv, sas7bdat
col.id	A name of column containing subject ID. This ID should be the same in both longdat and vitstat tables. If not provided, the first column in the x and y will be used by default.
col.status	A name of the column containing status variable (0/1 which indicate alive/dead). If not provided - then the column #2 from the vital statistics dataset will be used.
col.age	A name of age column (also called 't1'). If not provided then the 3rd column from the longitudinal dataset (x) will be used.
col.age.event	A name of 'event' column. The event column indicates a time when the even occured (e.g. system failure). If not provided then the 3rd column from the vital statistics dataset will be used.
covariates	A list of covariates. If covariates not provided, then all columns from longitudinal table having index > 3 will be used as covariates.
interval	A number of breaks between observations for discrete model. Default = 1 unit of time.
verbose	A verbosing output indicator. Default=FALSE.

Value

A list of two elements: first element contains a preprocessed data for continuous model, with arbitrary intervals between observations and second element contains a preprocessed data table for a discrete model (with constant intervals between observations).

Examples

```
library(spm)
data <- prepare_data(x=system.file("data", "longdat.csv", package="spm"), y=system.file("data", "vitstat.csv",
  head(data[[1]])
  head(data[[2]])
```

simdata_cont

*Multi-dimensional simulation function for continuous trait.***Description**

Multi-dimensional simulation function for continuous trait.

Usage

```
simdata_cont(N = 100, a = -0.05, f1 = 80, Q = 2e-07, f = 80, b = 5,
  mu0 = 2e-05, theta = 0.08, step = 0.05, tstart = 30, tend = 105,
  ystart = 80, sd0 = 4, k = 1)
```

Arguments

N	Number of individuals.
a	A k by k matrix, which characterize the rate of the adaptive response.
f1	A particular state, which if a deviation from the normal (or optimal). This is a vector with length of k.
Q	A matrix k by k, which is a non-negative-definite symmetric matrix.
f	A vector-function (with length k) of the normal (or optimal) state.
b	A diffusion coefficient, k by k matrix.
mu0	mortality at start period of time.
theta	A displacement coefficient of the Gompertz function.
tstart	A number that defines starting time (30 by default).
tend	A number, defines final time (105 by default).
ystart	A vector with length equal to number of dimensions used, defines starting values of covariates.
k	number of dimensions (k = 1 by default).

Value

A table with simulated data.

Examples

```
library(spm)
dat <- simdata_cont(N=500)
dat
```

simdata_cont_1D	<i>One-dimensional simulation function for continuous time (arbitrary intervals between observations).</i>
-----------------	--

Description

One-dimensional simulation function for continuous time (arbitrary intervals between observations).

Usage

```
simdata_cont_1D(N = 10, aH = -0.05, f1H = 80, QH = 2e-07, fH = 80,
  bH = 5, mu0H = 2e-05, thetaH = 0.08, step = 0.05, tstart = 30,
  tend = 105, ystart = 80, sd0 = 4)
```

Arguments

N	Number of individuals.
tstart	A number that defines starting time (30 by default).
tend	A number, defines final time (105 by default).
ystart	A vector with length equal to number of dimensions used, defines starting values of covariates.
a	A k by k matrix, which characterize the rate of the adaptive response.
f1	A particular state, which if a deviation from the normal (or optimal). This is a vector with length of k.
Q	A matrix k by k, which is a non-negative-definite symmetric matrix.
f	A vector-function (with length k) of the normal (or optimal) state.
b	A diffusion coefficient, k by k matrix.
mu0	mortality at start period of time.
theta	A displacement coefficient of the Gompertz function.

Value

A table with simulated data.

Examples

```
library(spm)
dat <- simdata_cont_1D(N=2500)
dat
```

simdata_discr

*Multi-dimension simulation function***Description**

Multi-dimension simulation function

Usage

```
simdata_discr(N = 100, a = -0.05, f1 = 80, Q = 2e-08, f = 80, b = 5,
  mu0 = 1e-05, theta = 0.08, ystart = 80, tstart = 30, tend = 105,
  dt = 1, k = 1)
```

Arguments

N	Number of individuals
a	A k by k matrix, which characterize the rate of the adaptive response.
f1	A particular state, which is a deviation from the normal (or optimal). This is a vector with length of k.
Q	A matrix k by k, which is a non-negative-definite symmetric matrix.
f	A vector-function (with length k) of the normal (or optimal) state.
b	A diffusion coefficient, k by k matrix.
mu0	mortality at start period of time.
theta	A displacement coefficient of the Gompertz function.
ystart	A vector with length equal to number of dimensions used, defines starting values of covariates.
tstart	A number that defines starting time (30 by default).
tend	A number, defines final time (105 by default).
dt	A time step (1 by default).
k	number of dimensions (k = 1 by default).

Value

A table with simulated data.

Examples

```
library(spm)
data <- simdata_discr(N=1000, ystart=80, k=1)
head(data)
```

<code>simdata_time_dep</code>	<i>Simulation function for continuous trait with time-dependant coefficients.</i>
-------------------------------	---

Description

Simulation function for continuous trait with time-dependant coefficients.

Usage

```
simdata_time_dep(N = 10, formulas = list(at = "-0.05", f1t = "80", Qt =
  "2e-7*exp(0.08*t)", ft = "80", bt = "5", mu0t = "2e-5*exp(0.08*t)"),
  step = 0.05, tstart = 30, tend = 105, ystart = 80, sd0 = 4, k = 1)
```

Arguments

<code>N</code>	Number of individuals.
<code>formulas</code>	: a list of formulas that define age (time) - dependency. Default: <code>list(at="a", f1t="f1", Qt="Q*exp(theta*t)", ft="f", bt="b", mu0t="mu0*exp(theta*t)")</code>
<code>tstart</code>	A number that defines starting time (30 by default).
<code>tend</code>	A number, defines final time (105 by default).
<code>ystart</code>	A starting value of covariates.

Value

A table with simulated data.

Examples

```
library(spm)
dat <- simdata_time_dep(N=2500)
dat
```

<code>spm</code>	<i>A central function that estimates Stochastic Process Model (SPM) parameters a from given dataset.</i>
------------------	--

Description

A central function that estimates Stochastic Process Model (SPM) parameters a from given dataset.

Usage

```
spm(x, model = "discrete", formulas = NULL, verbose = FALSE, tol = NULL)
```

Arguments

x	A dataset: is the output from prepare_data(...) function and consists of two separate data tables: (1) a data table for continuous-time model and (2) a data table for discrete-time model.
model	A model type. Choices are: "discrete", "continuous" or "time-dependent".
formulas	A list of parameter formulas used in the "time-dependent" model.
verbose	A verbosing output indicator (FALSE by default).
tol	A tolerance threshold for matrix inversion (NULL by default).

Value

For "discrete" and "continuous" model types: (1) a list of model parameter estimates for the discrete model type described in "", Akushevich et al, 2005 and (2) a list of model parameter estimates for the continuous model type described in "", Yashin et al, 2007.

For the "time-dependent" model (model parameters depend on time): a set of model parameter estimates.

Examples

```
library(spm)
#Prepare data for optimization
data <- prepare_data(x=system.file("data", "longdat.csv", package="spm"), y=system.file("data", "vitstat.csv"),
#Parameters estimation (default model: discrete-time):
p.discr.model <- spm(data)
p.discr.model
# Continuous-time model:
p.cont.model <- spm(data, model="continuous")
p.cont.model
# Model with time-dependent coefficients:
data <- prepare_data(x=system.file("data", "longdat.csv", package="spm"), y=system.file("data", "vitstat.csv"),
p.td.model <- spm(data, model="time-dependent")
p.td.model
```

spm_continuous

Continuous multi-dimensional optimization

Description

Continuous multi-dimensional optimization

Usage

```
spm_continuous(dat, a = 0.05, f1 = 80, Q = 2e-08, f = 81, b = 5,
mu0 = 2e-05, theta = 0.08, k = 1, verbose = F)
```

Arguments

dat	A data table.
a	A starting value of the rate of adaptive response to any deviation of Y from f1(t).
f1	A starting value of the average age trajectories of the variables which process is forced to follow.
Q	Starting values of the quadratic hazard term.
f	A starting value of the "optimal" value of variable which corresponds to the minimum of hazard rate at a respective time.
b	A starting value of a diffusion coefficient representing a strength of the random disturbance from Wiener Process.
mu0	A starting value of the baseline hazard.
theta	A starting value of the parameter theta (axe displacement of Gompertz function).
k	A number of dimensions.
verbose	An indicator of verbosing output.
tol	A tolerance threshold for matrix inversion.

Details

spm_integral_MD runs much slower than discrete but more precise and can handle time intervals with different lengths.

Value

A set of estimated parameters a, f1, Q, f, b, mu0, theta.

Examples

```
library(spm)
# Reading the data:
longdat <- read.csv(system.file("data", "longdat.csv", package="spm"))
vitstat <- read.csv(system.file("data", "vitstat.csv", package="spm"))
dd <- prepare_data(longdat=longdat, vitstat=vitstat, interval=1, col.status="IsDead", col.id="ID", col.age="Age")
data <- dd[[1]][,2:6]
#Parameters estimation:
pars <- spm_continuous(dat=data, a=-0.05, f1=80, Q=2e-8, f=80, b=5, mu0=2e-5, theta=0.08, k=1)
pars
```

spm_continuous_1D

Continuous one-dimensional optimization

Description

Continuous one-dimensional optimization

Usage

```
spm_continuous_1D(dat, a = -0.05, f1 = 80, Q = 2e-08, f = 80, b = 5,
  mu0 = 2e-05, theta = 0.08)
```


Arguments

dat	A data table.
a	A starting value of the rate of adaptive response to any deviation of Y from f1(t).
f1	A starting value of the average age trajectories of the variables which process is forced to follow.
Q	Starting values of the quadratic hazard term.
f	A starting value of the "optimal" value of variable which corresponds to the minimum of hazard rate at a respective time.
b	A starting value of a diffusion coefficient representing a strength of the random disturbance from Wiener Process.
mu0	A starting value of the baseline hazard.
theta	A starting value of the parameter theta (axe displacement of Gompertz function).
k	A number of dimensions.
verbose	An indicator of verbosing output.
tol	A tolerance threshold for matrix inversion.

Details

spm_integral_1D runs much slower than discrete but more precise and can handle time intervals with different lengths.

Value

A set of estimated parameters a, f1, Q, f, b, mu0, theta.

Examples

```
library(spm)
# Reading the data:
longdat <- read.csv(system.file("data", "longdat.csv", package="spm"))
vitstat <- read.csv(system.file("data", "vitstat.csv", package="spm"))
dd <- prepare_data(longdat=longdat, vitstat=vitstat, interval=1, col.status="IsDead", col.id="ID", col.age="Age")
data <- dd[[1]][,2:6]
#Parameters estimation:
pars <- spm_continuous_1D(dat=data, a=-0.05, f1=80, Q=2e-8, f=80, b=5, mu0=2e-5, theta=0.08)
pars
```

spm_discrete

Discrete multi-dimensional optimization

Description

Discrete multi-dimensional optimization

Usage

```
spm_discrete(dat, k = 1, theta_range = seq(0.001, 0.09, by = 0.001),
  tol = NULL)
```

Arguments

<code>dat</code>	A data table.
<code>k</code>	A number of dimensions.
<code>theta_range</code>	A range of theta parameter (axe displacement of Gompertz function), default: from 0.001 to 0.09 with step of 0.001.
<code>tol</code>	A tolerance threshold for matrix inversion (NULL by default).

Details

This function is way much faster than continuous `spm_continuous_MD(...)` (but less precise) and used mainly in estimation as a starting point for the `spm_continuous_MD(...)`.

Value

A list of two elements: (1) estimated parameters `u`, `R`, `b`, `epsilon`, `Q`, `mu0`, `theta` and (2) estimated parameters `a`, `f1`, `Q`, `f`, `b`, `mu0`, `theta`. Note: `b` and `mu0` from first list are different from `b` and `mu0` from the second list.

Examples

```
library(spm)
# Reading longitudinal data
longdat <- read.csv(system.file("data", "longdat.csv", package="spm"))
# Prepare data for optimization
vitstat <- read.csv(system.file("data", "vitstat.csv", package="spm"))
data <- prepare_data(longdat=longdat, vitstat=vitstat, interval=1, col.status="IsDead", col.id="ID", col.age=
# Parameters estimation
pars <- spm_discrete(data[[2]], k=1, theta_range=seq(0.001, 0.09, by=0.001), tol=NULL)
pars
```

<code>spm_time_dep</code>	<i>spm_time_dep : a function that can handle time-dependant coefficients:</i>
---------------------------	---

Description

`spm_time_dep` : a function that can handle time-dependant coefficients:

Usage

```
spm_time_dep(x, start = list(a = -0.5, f1 = 80, Q = 2e-08, f = 80, b = 5, mu0 = 1e-05), formulas = list(at = "a", f1t = "f1", Qt = "Q", ft = "f", bt = "b", mu0t = "mu0"), verbose = TRUE, lower_bound = NULL, upper_bound = NULL, factr = 1e-16, lmult = 0.5, umult = 2)
```

Arguments

<code>x</code>	: input data table.
<code>start</code>	: a list of starting parameters, default: <code>l1st(a=-0.5, f1=80, Q=2e-8, f=80, b=5, mu0=1e-5)</code> ,
<code>formulas</code>	: a list of formulas that define age (time) - dependency. Default: <code>list(at="a", f1t="f1", Qt="Q", ft="f", bt="b", mu0t="mu0")</code>

Value

optimal coefficients

Examples

```
library(spm)
#Data preparation:
n <- 1000
data <- simdata_time_dep(N=n)
opt.par <- spm_time_dep(data[,2:6])
opt.par
```

Index

`prepare_data`, [2](#)

`simdata_cont`, [3](#)

`simdata_cont_1D`, [4](#)

`simdata_discr`, [5](#)

`simdata_time_dep`, [6](#)

`spm`, [6](#)

`spm_continuous`, [7](#)

`spm_continuous_1D`, [8](#)

`spm_discrete`, [9](#)

`spm_time_dep`, [10](#)