Package ‘spm’

December 20, 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` *Data pre-processing for analysis with stochastic process model methodology.*

Description

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

Usage

```
prepare_data(longdat, vitstat, interval = 1, col.status = "IsDead",
  col.id = "ID", col.age = "Age", col.age.event = "LSmort",
  covariates = c("DBP", "BMI", "DBP1", "DBP2", "Weight", "Height"),
  verbose = TRUE)
```

Arguments

`longdat`	A table with longitude records.
`vitstat`	A table with vital statistics (mortality).
`interval`	A number of breaks between observations for discrete simulation. Default = 1 (no breaks).
`col.status`	A name of column containing status variable (0/1 which indicate alive/dead).
`col.id`	A name of column containing patient ID. This ID should be the same in both longdat and vitstat tables.
`col.age`	A name of age column.
`col.age.event`	A name of event column.
`covariates`	A list of covariates.
`verbose`	A verbosing output indicator. Default=TRUE.

Value

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

Examples

```
library(spm)
#Reading longitude data:
ldat <- read.csv(system.file("data","longdat.csv",package="spm"))
# Prepare data for optimization:
vdat <- read.csv(system.file("data","vitstat.csv",package="spm"))
data <- prepare_data(longdat=ldat, vitstat=vdat,interval=1, col.status="IsDead", col.id="ID", col.age="Age",
# Parameters estimation:
pars <- spm(data,k = 1)
pars
```

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

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)
```

`simdata_time_dep` *Simulation function for continuous trait with time-dependant coefficients.*

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

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

Value

A table with simulated data.

Examples

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

`spm` *Stochastic Process Modelling (SPM) A main function that estimates parameters a, f1, Q, f, b, mu0, theta from given dataset.*

Description

Stochastic Process Modelling (SPM) A main function that estimates parameters a, f1, Q, f, b, mu0, theta from given dataset.

Usage

```
spm(dat, k = 2, verbose = F, tol = NULL)
```

Arguments

dat	A dataset.
k	Number of dimensions.
verbose	A verbosing output indicator.
tol	A tolerance threshold for matrix inversion.

Value

A list of (1) Estimated starting point (from quick discrete optimization) and (2) Estimated coefficients.

Examples

```
library(spm)
#Prepare data for optimization
longdat <- read.csv(system.file("data","longdat.csv",package="spm"))
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="A
#Parameters estimation:
pars=spm(data,k = 1)
pars
```

---

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 that 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="A
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 that 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="A
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

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

Details

This function is way much faster that continuous `spm_continuous_MD(...)` (but less precise) and used mainly in estimation 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
```

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

Description

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

Usage

```
spm_time_dep(data, start = list(a1 = -0.5, a2 = 0.2, f1 = 80, Q = 2e-08, f =
  80, b = 5, mu0 = 1e-05, theta = 0.08), formulas = list(at = "a1*t+a2", f1t =
  "f1", Qt = "Q*exp(theta*t)", ft = "f", bt = "b", mu0t = "mu0*exp(theta*t)"),
  verbose = TRUE, lower_bound = NULL, upper_bound = NULL, factr = 1e-16,
  lmult = 0.5, umult = 2)
```

Arguments

`start`	: a list of starting parameters, default: llist(a=-0.5, f1=80, Q=2e-8, f=80, b=5, mu0=1e-5, theta=0.08),
`formulas`	: a list of formulas that define age (time) - dependency. Default: list(at="a1*t+a2", f1t="f1", Qt="Q*exp(theta*t)", ft="f", bt="b", mu0t="mu0*exp(theta*t)")

Value

optimal coefficients

Examples

```
library(spm)
#Data preparation:
N <- 1000
data <- simdata_time_dep(N=2500)
opt.par <- spm_time_dep(data[,2:6], formulas=list(at="a", f1t="f1", Qt="Q*exp(theta*t)", ft="f", bt="b", mu0
opt.par
```

Index