Package ‘stpm’

June 1, 2022

Type     Package
Title    Stochastic Process Model for Analysis of Longitudinal and
          Time-to-Event Outcomes
Version  1.7.11
Date     2022-05-25
Author   I. Zhbannikov, Liang He, K. Arbeev, I. Akushevich, A. Yashin.
Maintainer Ilya Y. Zhbannikov <ilya.zhbannikov@duke.edu>
Description Utilities to estimate parameters of the models with survival functions
    induced by stochastic covariates. Miscellaneous functions for data preparation
    and simulation are also provided. For more information, see:
    (i) "Stochastic model for analysis of longitudinal data on aging and mortality"
    by Yashin A. et al. (2007),
    Mathematical Biosciences, 208(2), 538-551, <DOI:10.1016/j.mbs.2006.11.006>;
    (ii) "Health decline, aging and mortality: how are they related?"
    by Yashin A. et al. (2007),
License  GPL
Imports  sas7bdat, stats, nloptr, survival, tools, MASS
LinkingTo Rcpp, RcppArmadillo
Depends  R (>= 2.10), Rcpp (>= 0.11.1)
VignetteBuilder knitr
Suggests knitr (>= 1.11), rmarkdown (>= 1.9)
RoxygenNote 7.1.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2022-05-31 23:50:05 UTC

R topics documented:

  ex_data ................................................................................... 3
R topics documented:

- func1 ................................................................. 3
- get.column.index .................................................. 4
- getNextY.cont ..................................................... 4
- getNextY.cont2 ..................................................... 5
- getNextY.discr ..................................................... 5
- getNextY.discr.m ................................................... 6
- getPrevY.discr ..................................................... 6
- getPrevY.discr.m ................................................... 7
- longdat ............................................................. 7
- LRTTest ............................................................. 8
- m ................................................................. 8
- make.short.format ................................................ 9
- mu ............................................................... 9
- prepare_data ....................................................... 10
- prepare_data_cont ............................................... 11
- prepare_data_discr .............................................. 12
- sigma_sq .......................................................... 13
- simdata_cont ..................................................... 13
- simdata_discr ..................................................... 15
- simdata_gamma_frailty ........................................... 16
- simdata_time_dep ............................................... 17
- sim_pobs .......................................................... 19
- spm ............................................................ 21
- spm.impute ........................................................ 23
- spm_continuous .................................................. 24
- spm_cont_lin ...................................................... 26
- spm_cont_quad_lin ................................................ 28
- spm_con_1d ........................................................ 30
- spm_con_1d_g ...................................................... 32
- spm_discrete ....................................................... 34
- spm_pobs .......................................................... 35
- spm_projection .................................................... 37
- spm_time_dep ...................................................... 39
- stpm ............................................................. 40
- trim ............................................................. 41
- trim.leading ....................................................... 42
- trim.trailing ....................................................... 42
- vitstat ............................................................. 42

Index ............................................................... 43
This is the longitudinal genetic dataset.

Author(s)
Liang He

An internal function to compute m and gamma based on continuous-time model (Yashin et. al., 2007)

Usage
func1(tt, y, a, f1, Q, f, b, theta)

Arguments
- tt: tt - time
- y
- a: a (see Yashin et. al, 2007)
- f1: f1 (see Yashin et. al, 2007)
- Q: Q (see Yashin et. al, 2007)
- f: f (see Yashin et. al, 2007)
- b: b (see Yashin et. al, 2007)
- theta

Value
list(m, gamma) Next values of m and gamma (see Yashin et. al, 2007)
get.column.index  
An internal function to obtain column index by its name

Description
An internal function to obtain column index by its name

Usage
get.column.index(x, col.name)

Arguments
x  
Dataset
col.name  
Column name

Value
column index(es) in the provided dataset

getNextY.cont  
An internal function to compute next Y based on continous-time model (Yashin et. al., 2007)

Description
An internal function to compute next Y based on continous-time model (Yashin et. al., 2007)

Usage
getNextY.cont(y1, t1, t2, a, f1, Q, f, b, theta)

Arguments
y1  
y1
t1  
t1
t2  
t2
a  
a (see Yashin et. al, 2007)
f1  
f1 (see Yashin et. al, 2007)
Q  
Q (see Yashin et. al, 2007)
f  
f (see Yashin et. al, 2007)
b  
b (see Yashin et. al, 2007)
theta  
theta (see Yashin et. al, 2007)
getNextY.cont2

Value

y.next Next value of Y

Description

An internal function to compute next value of physiological variable Y

Usage

getNextY.cont2(y1, t1, t2, b, a, f1)

Arguments

y1

 t1

 t2

 b b (see Yashin et. al, 2007)

 a a (see Yashin et. al, 2007)

 f1 f1 (see Yashin et. al, 2007)

Value

y.next Next value of y

getNextY.discr

An internal function to compute the next value of physiological variable Y based on discrete-time model (Akushevich et. al., 2005)

Description

An internal function to compute the next value of physiological variable Y based on discrete-time model (Akushevich et. al., 2005)

Usage

getNextY.discr(y1, u, R, Sigma)
Arguments

<table>
<thead>
<tr>
<th>y1</th>
<th>y1</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>u (see Akushevich et. al, 2005)</td>
</tr>
<tr>
<td>R</td>
<td>R (see Akushevich et. al, 2005)</td>
</tr>
<tr>
<td>Sigma</td>
<td>Sigma (see Akushevich et. al, 2005)</td>
</tr>
</tbody>
</table>

Value

y.next Next value of y

---

getNextY.discr.m  An internal function to compute next m based on discrete-time model

Description

An internal function to compute next m based on discrete-time model

Usage

g getNextY.discr.m(y1, u, R)

Arguments

<table>
<thead>
<tr>
<th>y1</th>
<th>y1</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>

Value

m Next value of m (see Yashin et. al, 2007)

---

getPrevY.discr  An internal function to compute previous value of physiological variable Y based on discrete-time model

Description

An internal function to compute previous value of physiological variable Y based on discrete-time model

Usage

g getPrevY.discr(y2, u, R, Sigma)
Arguments

y2       y2
u        u
R        R
Sigma    Sigma

Value

y1 Previous value of y

getPrevY.discr.m  An internal function to compute previous m based on discrete-time model

Description

An internal function to compute previous m based on discrete-time model

Usage

getPrevY.discr.m(y2, u, R)

Arguments

y2       y2
u        u
R        R

Value

m Next value of m (see Yashin et. al, 2007)

longdat  This is the longitudinal dataset.

Description

This is the longitudinal dataset.

Author(s)

Ilya Y Zhbannikov <ilya.zhbannikov@duke.edu>
**LRTest**  
*Likelihood-ratio test*

**Description**
Likelihood-ratio test

**Usage**

LRTest(LA, L0, df = 1)

**Arguments**
- **LA**  
  Log-likelihood for alternative hypothesis
- **L0**  
  Log-likelihood for null hypothesis
- **df**  
  Degrees of freedom for Chi-square test

**Value**

p-value of LR test.

---

**m**  
*An internal function to compute m from*

**Description**
An internal function to compute m from

**Usage**

m(y, t1, t2, a, f1)

**Arguments**
- **y**  
  Current value of Y
- **t1**  
  t1
- **t2**  
  t2
- **a**  
  a (see Yashin et. al, 2007)
- **f1**  
  f1 (see Yashin et. al, 2007)

**Value**

m m (see Yashin et. al, 2007)
### make.short.format

An internal function which construct short data format from a given long

**Usage**

```r
make.short.format(
  x,
  col.id = 1,
  col.status = 2,
  col.t1 = 3,
  col.t2 = 4,
  col.cov = 5
)
```

**Arguments**

- `x`: Dataset
- `col.id`: Column ID index
- `col.status`: Column status index
- `col.t1`: Column t1 index
- `col.t2`: Column t2 index
- `col.cov`: Column covariates indices

**Value**

column index(es) in the provided dataset

### mu

An internal function to compute mu

**Description**

An internal function to compute mu

**Usage**

```r
mu(y, mu0, b, Q, theta, tt)
```
Arguments

y  Current value of y
mu0  mu0 (see Yashin et. al, 2007)
b  b (see Yashin et. al, 2007)
Q  Q (see Yashin et. al, 2007)
theta  theta (see Yashin et. al, 2007)
tt  t (time)

Value

mu  Next value of mu

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(
  x,
  col.id = NA,
  col.status = NA,
  col.age = NA,
  col.age.event = NA,
  covariates = NA,
  interval = 1,
  verbose = FALSE
)

Arguments

x  A path to the file with table of follow-up oservations (longitudinal table). File formats: csv, sas7bdat

col.id  A name of column containing subject ID. This ID should be the same in both x (longitudinal) and y (vital statistics) tables. None: if col.id not provided, the first column of the x and first column of the y will be used by default.

col.status  A name of the column containing status variable (0/1, which is an indicator of death/censoring). Note: if not provided - then the column #2 from the y (vital statistics) dataset will be used.
prepare_data_cont

Description

Prepares continuous-time dataset.

Usage

prepare_data_cont(
  merged.data,  
  col.status.ind,  
  col.id.ind,  
  col.age.ind,  
  col.age.event.ind,  
  col.covar.ind,  
  verbose,  
  dt
)

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

## Not run:
library(stpm)

data <- prepare_data(x=system.file("extdata","longdat.csv",package="stpm"))
head(data[[1]])
head(data[[2]])

## End(Not run)
prepare_data_discr

Arguments
merged.data a longitudinal study dataset.
col.status.ind index of "status" column.
col.id.ind subject id column index.
col.age.ind index of the age column.
col.age.event.ind an index of the column which represents the time in which event occured.
col.covar.ind a set of column indexes which represent covariates.
verbose turns on/off verbosing output.
dt interval between observations.

prepare_data_discr Prepares discrete-time dataset.

Description
Prepares discrete-time dataset.

Usage
prepare_data_discr(
  merged.data,
  interval,
  col.status.ind,
  col.id.ind,
  col.age.ind,
  col.age.event.ind,
  col.covar.ind,
  verbose
)

Arguments
merged.data a longitudinal study dataset.
interval interval between observations.
col.status.ind index of "status" column.
col.id.ind subject id column index.
col.age.ind index of the age column.
col.age.event.ind an index of the column which represents the time in which event occured.
col.covar.ind a set of column indexes which represent covariates.
verbose turns on/off verbosing output.
**sigma_sq**

An internal function to compute sigma square analytically

**Description**

An internal function to compute sigma square analytically

**Usage**

`sigma_sq(t1, t2, b)`

**Arguments**

- `t1`  
- `t2`  
- `b` (see Yashin et. al, 2007)

**Value**

`sigma_square` (see Akushevich et. al, 2005)

---

**simdata_cont**

Multi-dimensional simulation function for continuous-time SPM.

**Description**

Multi-dimensional simulation function for continuous-time SPM.

**Usage**

```r
simdata_cont(
  N = 10,
  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,
  sd0 = 1,
  nobs = NULL,
)```

```
gomp = TRUE,
    format = "long"
)

Arguments

N  Number of individuals.
a  A k by k matrix, represents the adaptive capacity of the organism
f1 A trajectory that corresponds to the long-term average value of the stochastic
    process Y(t), which describes a trajectory of individual covariate (physiological
    variable) influenced by different factors represented by a random Wiener process
    W(t). This is a vector with length of k.
Q  A matrix k by k, which is a non-negative-definite symmetric matrix, represents
    a sensitivity of risk function to deviation from the norm.
f  A vector with length of k, represents the normal (or optimal) state of physiolog-
    ical variable.
b  A diffusion coefficient, k by k matrix, characterizes a strength of the random
    disturbances from Wiener process W(t).
mu0 A baseline mortality.
theta A displacement coefficient.
ystart A vector with length equal of k, 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 discrete step size between two observations. A random uniform value is then
    added to this step size.
sdθ a standard deviation for modelling the next covariate value.
nobs A number of observations (lines) for individual observations.
gomp A flag (FALSE by default). When it is set, then time-dependent exponential
    form of mu0 and Q are used: mu0 = mu0*exp(theta*t).
format Data format: "long" (default), "short".

Value

A table with simulated data.

References


Examples

library(stpm)
dat <- simdata_cont(N=50)
head(dat)
**simdata_discr**  

*Multi-dimension simulation function*

**Description**

Multi-dimension simulation function

**Usage**

```r
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,
  nobs = NULL,
  format = "long"
)
```

**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. Default ystart = 80.  
- **tstart**: Starting time (age). Can be a number (30 by default) or a vector of two numbers: c(a,b) - in this case, starting value of time is simulated via uniform(a,b) distribution.  
- **tend**: A number, defines final time (105 by default).  
- **dt**: A time step (1 by default).
simdata_gamma_frailty

   nobs       A number, defines a number of observations (lines) for an individual, NULL by default.
   format     Data format: "long" (default), "short".

Value

   A table with simulated data.

References


Examples

   library(stpm)
   data <- simdata_discr(N=100)
   head(data)

simdata_gamma_frailty This script simulates data using familial frailty model.
   We use the following variation: gamma(mu, ssq),
   where mu is the mean and ssq is sigma square. See:

Description

   This script simulates data using familial frailty model. We use the following variation: gamma(mu, ssq), where mu is the mean and ssq is sigma square. See: https://www.rocscience.com/help/swedge/webhelp/swedge/Gamma_Distribution.htm

Usage

   simdata_gamma_frailty(
     N = 10,
     f = list(at = "-0.05", f1t = "80", Qt = "2e-8", ft = "80", bt = "5", mu0t = "1e-3"),
     step = 1,
     tstart = 30,
     tend = 105,
     ystart = 80,
     sd0 = 1,
     nobs = NULL,
     gamma_mu = 1,
     gamma_ssq = 0.5
   )
**Arguments**

- **N**  Number of individuals.
- **f**  A list of formulas that define age (time) - dependency. Default: list(at="a", Qf1t="Q*exp(theta*t)", ft="f"t", bt="b", mu0t="mu0*exp(theta*t)")
- **step**  An interval between two observations, a random uniformly-distributed value is then added to this step.
- **tstart**  Starting time (age). Can be a number (30 by default) or a vector of two numbers: c(a, b) - in this case, starting value of time is simulated via uniform(a,b) distribution.
- **tend**  A number, defines final time (105 by default).
- **ystart**  A starting value of covariates.
- **sd0**  A standard deviation for modelling the next covariate value, sd0 = 1 by default.
- **nobs**  A number of observations (lines) for individual observations.
- **gamma_mu**  A parameter which is a mean value, default = 1
- **gamma_ssq**  A sigma squared, default = 0.5.

**Value**

A table with simulated data.

**References**


**Examples**

```r
library(stpm)
dat <- simdata_gamma_frailty(N=10)
head(dat)
```

---

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

**Description**

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

Usage

simdata_time_dep(
  N = 10,
  f = list(at = "-0.05", f1t = "80", Qt = "2e-8", ft = "80", bt = "5", mu0t = "1e-3"),
  step = 1,
  tstart = 30,
  tend = 105,
  ystart = 80,
  sd0 = 1,
  nobs = NULL,
  format = "short"
)

Arguments

N
  Number of individuals.

f
  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)"

step
  An interval between two observations, a random uniformly-distributed value
  is then added to this step.

tstart
  Starting time (age). Can be a number (30 by default) or a vector of two num-
  bers: c(a, b) - in this case, starting value of time is simulated via uniform(a,b)
  distribution.

tend
  A number, defines final time (105 by default).

ystart
  A starting value of covariates.

sd0
  A standard deviation for modelling the next covariate value, sd0 = 1 by default.

nobs
  A number of observations (lines) for individual observations.

format
  Data format: "short" (default), "long".

Value

A table with simulated data.

References

Yashin, A. et al (2007), Health decline, aging and mortality: how are they related? Biogerontology,

Examples

library(stpm)
dat <- simdata_time_dep(N=100)
head(dat)
**Description**

Multi-dimension simulation function for data with partially observed covariates (multidimensional GenSPM) with arbitrary intervals

**Usage**

```r
sim_pobs(
  N = 10,
  aH = -0.05,
  aL = -0.01,
  f1H = 60,
  f1L = 80,
  QH = 2e-08,
  QL = 2.5e-08,
  fH = 60,
  fL = 80,
  bH = 4,
  bL = 5,
  mu0H = 8e-06,
  mu0L = 1e-05,
  thetaH = 0.08,
  thetaL = 0.1,
  p = 0.25,
  ystart = 80,
  tstart = 30,
  tend = 105,
  dt = 1,
  sd0 = 1,
  mode = "observed",
  gomp = FALSE,
  nobs = NULL
)
```

**Arguments**

- **N**: Number of individuals.
- **aH**: A k by k matrix, which characterize the rate of the adaptive response when Z = 1.
- **aL**: A k by k matrix, which characterize the rate of the adaptive response when Z = 0.
f1H  A particular state, which if a deviation from the normal (or optimal) when Z = 1. This is a vector with length of k.
f1L  A particular state, which if a deviation from the normal (or optimal) when Z = 0. This is a vector with length of k.
QH  A matrix k by k, which is a non-negative-definite symmetric matrix when Z = 1.
QL  A matrix k by k, which is a non-negative-definite symmetric matrix when Z = 0.
fH  A vector-function (with length k) of the normal (or optimal) state when Z = 1.
fL  A vector-function (with length k) of the normal (or optimal) state when Z = 0.
bH  A diffusion coefficient, k by k matrix when Z = 1.
bL  A diffusion coefficient, k by k matrix when Z = 0.
mu0H  mortality at start period of time when Z = 1.
mu0L  mortality at start period of time when Z = 0.
thetaH  A displacement coefficient of the Gompertz function when Z = 1.
thetaL  A displacement coefficient of the Gompertz function when Z = 0.
p  A proportion of carriers in a simulated population (default p = 0.25).
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 discrete step size between two observations. A random uniform value is then added to this step size.
sd0  A standard deviation for modelling the next physiological variable (covariate) value.
mode  Can have the following values: "observed" (default), "unobserved". This represents a type of group to simulate: a group with observed variable Z, or group with unobserved variable Z.
gomp  A flag (FALSE by default). When it is set, then time-dependent exponential form of mu0 and Q are used: mu0 = mu0*exp(theta*t).
nobs  A number of observations (lines) for individual observations.

Value
A table with simulated data.

References

Examples
library(stpm)
dat <- sim_pobs(N=50)
head(dat)
A central function that estimates Stochastic Process Model parameters a from given dataset.

**Usage**

```r
spm(
  x,
  model = "discrete",
  formulas = list(at = "a", f1t = "f1", Qt = "Q", ft = "f", bt = "b", mu0t = "mu0"),
  start = NULL,
  tol = NULL,
  stopifbound = FALSE,
  lb = NULL,
  ub = NULL,
  pinv.tol = 0.01,
  theta.range = seq(0.01, 0.2, by = 0.001),
  verbose = FALSE,
  gomp = FALSE,
  opts = list(algorithm = "NLOPT_LN_NELDERMEAD", maxeval = 100, ftol_rel = 1e-08)
)
```

**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. Default: `formulas=list(at="a", f1t="f1", Qt="Q", ft="f", bt="b", mu0t="mu0")`.
- `start`: A starting values of coefficients in the "time-dependent" model.
- `tol`: A tolerance threshold for matrix inversion (NULL by default).
- `stopifbound`: A flag (default=FALSE) if it is set then the optimization stops when any of the parameters achieves lower or upper boundary.
- `lb`: Lower boundary, default NULL.
- `ub`: Upper boundary, default NULL.
- `pinv.tol`: A tolerance threshold for matrix pseudo-inverse. Default: 0.01.
- `theta.range`: A user-defined range of the parameter theta used in discrete-time optimization and estimating of starting point for continuous-time optimization.
- `verbose`: A verbosing output indicator (FALSE by default).
gomp

A flag (FALSE by default). When it is set, then time-dependent exponential
form of mu0 and Q are used: mu0 = mu0*exp(theta*t), Q = Q*exp(theta*t).

opts

A list of options for nloptr. Default value: opt=list(algorithm="NLOPT_LN_NELDERMEAD", maxeval=
ftol_rel=1e-8). Please see nloptr documentation for more information.

Value

For "discrete" (dmodel) and "continuous" (cmodel) model types: (1) a list of model parameter esti-
mates for the discrete model type described in "Life tables with covariates: Dynamic Model for
Nonlinear Analysis of Longitudinal Data", Akushevich et al, 2005.<DOI:10.1080/08898480590932296>,
and (2) a list of model parameter estimates for the continuous model type described in "Stochas-
tic model for analysis of longitudinal data on aging and mortality", Yashin et al, 2007, Math
Biosci.<DOI:10.1016/j.mbs.2006.11.006>.

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

References

Mathematical Biosciences, 208(2), 538-551.

for Nonlinear Analysis of Longitudinal Data. Mathematical Popu-
lation Studies, 12(2), pp.: 51-80. <DOI: 10.1080/08898480590932296>.

Yashin, A. et al (2007), Health decline, aging and mortality: how are they related? Biogerontology,

Examples

```r
## Not run:
library(stpm)
data.continuous <- simdata_cont(N=1000)
data.discrete <- simdata_discr(N=1000)
data <- list(data.continuous, data.discrete)
p.discr.model <- spm(data)
p.discr.model
p.cont.model <- spm(data, model="continuous")
p.cont.model
p.td.model <- spm(data, model="time-dependent", f=list(at="aa*t+bb", f1t="f1", Qt="Q", ft="f", bt="b", mu0t="mu0"),
start=list(a=-0.001, bb=0.05, f1=80, Q=2e-8, f=80, b=5, mu0=1e-3))
p.td.model

## End(Not run)
```
Description

Multiple Data Imputation with SPM

Usage

```r
spm.impute(
  x,
  id = 1,
  case = 2,
  t1 = 3,
  t2 = 3,
  covariates = 4,
  minp = 5,
  theta_range = seq(0.01, 0.2, by = 0.001)
)
```

Arguments

- **x**: A longitudinal dataset with missing observations
- **id**: A name (text) or index (numeric) of ID column. Default: 1
- **case**: A case status column name (text) or index (numeric). Default: 2
- **t1**: A t1 (or t if short format is used) column name (text) or index (numeric). Default: 3
- **t2**: A t2 column name (if long format is used) (text) or index (numeric). Default: 4
- **covariates**: A list of covariate column names or indices. Default: 5
- **minp**: Number of imputations. Default: 5
- **theta_range**: A range of parameter theta used for optimization, default: `seq(0.01, 0.15, by=0.001)`.

Value

- **imputed**: An imputed dataset.
- **imputations**: Temporary imputed datasets used in multiple imputations.

Examples

```r
## Not run:
library(stpm)
## Data preparation ##
data <- simdata_discr(N=1000, dt = 2)
miss.id <- sample(x=dim(data)[1], size=round(dim(data)[1]/4)) # ~25% missing data
```
incomplete.data <- data
incomplete.data[miss.id,5] <- NA
incomplete.data[miss.id-1,6] <- NA
## End of data preparation ##

# Estimate parameters from the complete dataset #
p <- spm_discrete(data, theta_range = seq(0.075, 0.09, by=0.001))
p

##### Multiple imputation with SPM ######
imp.data <- spm.impute(x=incomplete.data,
    minp=5,
    theta_range=seq(0.075, 0.09, by=0.001))$imputed
head(imp.data)
## Estimate SPM parameters from imputed data and compare them to the p ##
pp.test <- spm_discrete(imp.data, theta_range = seq(0.075, 0.09, by=0.001))
pp.test
## End(Not run)

---

**spm_continuous**  
*Continuous multi-dimensional optimization*

**Description**

Continuous multi-dimensional optimization

**Usage**

```r
spm_continuous(
  dat,
  a = -0.05,
  f1 = 80,
  Q = 2e-08,
  f = 80,
  b = 5,
  mu0 = 2e-05,
  theta = 0.08,
  stopifbound = FALSE,
  lb = NULL,
  ub = NULL,
  verbose = FALSE,
  pinv.tol = 0.01,
  gomp = FALSE,
  opts = list(algorithm = "NLOPT_LN_NELDERMEAD", maxeval = 100, ftol_rel = 1e-08),
  logmu0 = FALSE
)
```
**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>A data table.</td>
</tr>
<tr>
<td>a</td>
<td>A starting value of the rate of adaptive response to any deviation of Y from f1(t).</td>
</tr>
<tr>
<td>f1</td>
<td>A starting value of the average age trajectories of the variables which process is forced to follow.</td>
</tr>
<tr>
<td>Q</td>
<td>Starting values of the quadratic hazard term.</td>
</tr>
<tr>
<td>f</td>
<td>A starting value of the &quot;optimal&quot; value of variable which corresponds to the minimum of hazard rate at a respective time.</td>
</tr>
<tr>
<td>b</td>
<td>A starting value of a diffusion coefficient representing a strength of the random disturbance from Wiener Process.</td>
</tr>
<tr>
<td>mu0</td>
<td>A starting value of the baseline hazard.</td>
</tr>
<tr>
<td>theta</td>
<td>A starting value of the parameter theta (axe displacement of Gompertz function).</td>
</tr>
<tr>
<td>stopifbound</td>
<td>Estimation stops if at least one parameter achieves lower or upper boundaries.</td>
</tr>
<tr>
<td>lb</td>
<td>Lower bound of parameters under estimation.</td>
</tr>
<tr>
<td>ub</td>
<td>Upper bound of parameters under estimation. The program stops when the number of function evaluations exceeds maxeval. Default: 500.</td>
</tr>
<tr>
<td>verbose</td>
<td>An indicator of verbosing output.</td>
</tr>
<tr>
<td>gomp</td>
<td>A flag (FALSE by default). When it is set, then time-dependent exponential form of mu0 is used: mu0 = mu0<em>exp(theta</em>t).</td>
</tr>
<tr>
<td>opts</td>
<td>A list of options for nloptr. Default value: opt=list(algorithm=&quot;NLOPT_LN_NELDERMEAD&quot;,maxeval=100,ftol_rel=1e-8). Please see nloptr documentation for more information.</td>
</tr>
<tr>
<td>logmu0</td>
<td>Natural logarith of baseline mortality. Default: FALSE.</td>
</tr>
</tbody>
</table>

**Details**

`spm_continuous` 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 and additional variable `limit` which indicates if any parameter achieved lower or upper boundary conditions (FALSE by default).
- status Optimization status (see documentation for nloptr package).
- LogLik A logarithm likelihood.
- objective A value of objective function (given by nloptr).
- message A message given by nloptr optimization function (see documentation for nloptr package).
References


Examples

library(stpm)
set.seed(123)
#Reading the data:
data <- simdata_cont(N=2)
head(data)
#Parameters estimation:
pars <- spm_continuous(dat=data,a=-0.05, f1=80, Q=2e-8, f=80, b=5, mu0=2e-5)
pars

---

**spm_cont_lin**

*Continuous multi-dimensional optimization with linear terms in mu only*

**Description**

Continuous multi-dimensional optimization with linear terms in mu only

**Usage**

```r
spm_cont_lin(
dat,
a = -0.05,
f1 = 80,
Q = 2e-08,
f = 80,
b = 5,
mu0 = 2e-05,
theta = 0.08,
stopifbound = FALSE,
lb = NULL,
ub = NULL,
verbose = FALSE,
pinv.tol = 0.01,
gomp = FALSE,
opts = list(algorithm = "NLOPT_LN_NELDERMead", maxeval = 100, ftol_rel = 1e-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 linear 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).
stopifbound  Estimation stops if at least one parameter achieves lower or upper boundaries.
  #' Check the NLopt website for a description of the algorithms. Default: NLOPT_LN_NELDERMEAD
lb  Lower bound of parameters under estimation.
ub  Upper bound of parameters under estimation. The program stops when the number of function evaluations exceeds maxeval. Default: 500.
verbose  An indicator of verbosing output.
gomp  A flag (FALSE by default). When it is set, then time-dependent exponential form of mu0 is used: mu0 = mu0*exp(theta*t).
opts  A list of options for nloptr. Default value: opt=list(algorithm="NLOPT_LN_NELDERMEAD",maxeval=500,ftol_rel=1e-8). Please see nloptr documentation for more information.

Details

spm_continuous 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 and additional variable limit which indicates if any parameter achieved lower or upper boundary conditions (FALSE by default).
status Optimization status (see documentation for nloptr package).
LogLik A logarithm likelihood.
objective A value of objective function (given by nloptr).
message A message given by nloptr optimization function (see documentation for nloptr package).

References

Examples

```r
library(stpm)
set.seed(123)
#Reading the data:
data <- simdata_cont(N=2)
head(data)
#Parameters estimation:
pars <- spm_cont_lin(dat=data,a=-0.05, f1=80,
                     Q=2e-8, f=80, b=5, mu0=2e-5)
pars
```

### spm_cont_quad_lin

Continuous multi-dimensional optimization with quadratic and linear terms

#### Description

Continuous multi-dimensional optimization with quadratic and linear terms

#### Usage

```r
spm_cont_quad_lin(
  dat,
  a = -0.05,
  f1 = 80,
  Q = 2e-08,
  f = 80,
  b = 5,
  mu0 = 2e-05,
  theta = 0.08,
  Q1 = 1e-08,
  stopifbound = FALSE,
  lb = NULL,
  ub = NULL,
  verbose = FALSE,
  pinv.tol = 0.01,
  gomp = FALSE,
  opts = list(algorithm = "NLOPT_LN_NELDERMEAD", maxeval = 100, ftol_rel = 1e-08)
)
```

#### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dat</td>
<td>A data table.</td>
</tr>
<tr>
<td>a</td>
<td>A starting value of the rate of adaptive response to any deviation of Y from f1(t).</td>
</tr>
<tr>
<td>f1</td>
<td>A starting value of the average age trajectories of the variables which process is forced to follow.</td>
</tr>
</tbody>
</table>
Starting values of the quadratic hazard term.

A starting value of the "optimal" value of variable which corresponds to the minimum of hazard rate at a respective time.

A starting value of a diffusion coefficient representing a strength of the random disturbance from Wiener Process.

A starting value of the baseline hazard.

A starting value of the parameter theta (axe displacement of Gompertz function).

Q for linear term

Estimation stops if at least one parameter achieves lower or upper boundaries.

# Check the NLopt website for a description of the algorithms. Default: NLOPT_LN_NELDERMEAD

Lower bound of parameters under estimation.

Upper bound of parameters under estimation. The program stops when the number of function evaluations exceeds maxeval. Default: 500.

An indicator of verbosing output.


A flag (FALSE by default). When it is set, then time-dependent exponential form of mu0 is used: mu0 = mu0*exp(theta*t).

A list of options for nloptr. Default value: opt=list(algorithm="NLOPT_LN_NELDERMEAD",maxeval=100,ftol_rel=1e-8). Please see nloptr documentation for more information.

Details

spm_continuous 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 and additional variable limit which indicates if any parameter achieved lower or upper boundary conditions (FALSE by default).

Optimization status (see documentation for nloptr package).

A logarithm likelihood.

A value of objective function (given by nloptr).

A message given by nloptr optimization function (see documentation for nloptr package).

References

Examples

```r
library(stpm)
set.seed(123)
# Reading the data:
data <- simdata_cont(N=2)
head(data)
# Parameters estimation:
pars <- spm_con_quad_lin(dat=data, a=-0.05, f1=80,
                         Q=2e-8, f=80, b=5, mu0=2e-5, Q1=1e-08)
pars
```

---

**spm_con_1d**  
*Fitting a 1-D SPM model with constant parameters*

**Description**

This function implements an analytical solution to estimate the parameters in the continuous SPM model by assuming all the parameters are constants.

**Usage**

```r
spm_con_1d(
  spm_data,
  a = NA,
  b = NA,
  q = NA,
  f = NA,
  f1 = NA,
  mu0 = NA,
  theta = NA,
  lower = c(),
  upper = c(),
  control = list(xtol_rel = 1e-06),
  global = FALSE,
  verbose = TRUE,
  hessian = FALSE
)
```

**Arguments**

- `spm_data` A dataset for the SPM model. See the STPM package for more details about the format.
- `a` The initial value for the parameter $a$. The initial value will be predicted if not specified.
- `b` The initial value for the parameter $b$. The initial value will be predicted if not specified.
The initial value for the parameter \( q \). The initial value will be predicted if not specified.

The initial value for the parameter \( f \). The initial value will be predicted if not specified.

The initial value for the parameter \( f_1 \). The initial value will be predicted if not specified.

The initial value for the parameter \( \mu_0 \) in the baseline hazard. The initial value will be predicted if not specified.

The initial value for the parameter \( \theta \) in the baseline hazard. The initial value will be predicted if not specified.

A vector of the lower bound of the parameters.

A vector of the upper bound of the parameters.

A list of the control parameters for the optimization parameters.

A logical variable indicating whether the MLSL (TRUE) or the L-BFGS (FALSE) algorithm is used for the optimization.

A logical variable indicating whether initial information is printed.

A logical variable indicating whether the approximate (FALSE) or analytical (TRUE) Hessian is returned.

The estimates of the parameters.

The Hessian matrix of the estimates.

The minus log-likelihood.

A number indicating the convergence. See the `nloptr` package for more details.

Extra message about the convergence. See the `nloptr` package for more details.


Examples

```r
library(stpm)
dat <- simdata_cont(N=500)
colnames(dat) <- c("id", "x1", "t1", "t2", "y", "y.next")
res <- spm_con_1d(as.data.frame(dat), a=-0.05, b=2, q=1e-8, f=80, f1=90, mu0=1e-3, theta=0.08)
```
Fitting a 1-D genetic SPM model with constant parameters

Description

This function implements a continuous genetic SPM model by assuming all the parameters are constants.

Usage

```r
call_pm_con_1d_g(  
  spm_data,  
  gene_data,  
  a = NA,  
  b = NA,  
  q = NA,  
  f = NA,  
  f1 = NA,  
  mu0 = NA,  
  theta = NA,  
  effect = c("a"),  
  lower = c(),  
  upper = c(),  
  control = list(xtol_rel = 1e-06),  
  global = FALSE,  
  verbose = TRUE,  
  ahessian = FALSE,  
  method = "lbfgs",  
  method.hessian = "L-BFGS-B")
```

Arguments

- `spm_data`: A dataset for the SPM model. See the STPM pacakge for more details about the format.
- `gene_data`: A two column dataset containing the genotypes for the individuals in `spm_data`. The first column `id` is the ID of the individuals in `spm_data`, and the second column `geno` is the genotype.
- `a`: The initial value for the parameter `a`. The initial value will be predicted if not specified.
- `b`: The initial value for the parameter `b`. The initial value will be predicted if not specified.
- `q`: The initial value for the parameter `q`. The initial value will be predicted if not specified.
- `f`: The initial value for the parameter `f`. The initial value will be predicted if not specified.
The initial value for the parameter $f_1$. The initial value will be predicted if not specified.

mu0  
The initial value for the parameter $\mu_0$ in the baseline hazard. The initial value will be predicted if not specified.

theta  
The initial value for the parameter $\theta$ in the baseline hazard. The initial value will be predicted if not specified.

effect  
A character vector of the parameters that are linked to genotypes. The vector can contain any combination of $a$, $b$, $q$, $f$, $\mu_0$.

lower  
A vector of the lower bound of the parameters.

upper  
A vector of the upper bound of the parameters.

control  
A list of the control parameters for the optimization parameters.

global  
A logical variable indicating whether the MLSL (TRUE) or the L-BFGS (FALSE) algorithm is used for the optimization.

verbose  
A logical variable indicating whether initial information is printed.

ahessian  
A logical variable indicating whether the approximate (FALSE) or analytical (TRUE) Hessian is returned.

method  
Optimization method. Can be one of the following: lbfgs, mlsl, mma, slsqp, tnewton, varmetric. Default: lbfgs.

method.hessian  
Optimization method for hessian calculation (if ahessian=F). Default: L-BFGS-B.

Value

est  The estimates of the parameters.

hessian  The Hessian matrix of the estimates.

lik  The minus log-likelihood.

con  A number indicating the convergence. See the 'nloptr' package for more details.

message  Extra message about the convergence. See the 'nloptr' package for more details.

beta  The coefficients of the genetic effect on the parameters to be linked to genotypes.

References


Examples

# Not run:
library(stpm)
data(ex_SPMCon1dg)
res <- spm_con_1d_g(ex_data$spm_data, ex_data$gene_data,
a = -0.02, b=0.2, q=0.01, f=3, f1=3, mu0=0.01, theta=1e-05,
upper=c(-0.01,3,0.1,10,0.1,1e-05), lower=c(-1,0.01,0.00001,1,1,0.001,1e-05),
effect=c('q'))

# End(Not run)
Discrete multi-dimensional optimization

Description

Discrete multi-dimensional optimization

Usage

```r
spm_discrete(
  dat,
  theta_range = seq(0.02, 0.2, by = 0.001),
  tol = NULL,
  verbose = FALSE
)
```

Arguments

- `dat`: A data table.
- `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).
- `verbose`: An indicator of verbosing output.

Details

This function is way more faster than 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 ("dmodel", "cmodel"): (1) estimated parameters u, R, b, Sigma, Q, mu0, theta for discrete-time model and (2) estimated parameters a, f1, Q, f, b, mu0, theta for continuous-time model. Note: b and mu0 from first list are different from b and mu0 from the second list.

References

Examples

```r
library(stpm)
data <- simdata_discr(N=10)
#Parameters estimation
pars <- spm_discrete(data)
pars
```

---

**spm_pobs**

*Continuous-time multi-dimensional optimization for SPM with partially observed covariates (multidimensional GenSPM)*

---

**Description**

Continuous-time multi-dimensional optimization for SPM with partially observed covariates (multidimensional GenSPM)

**Usage**

```r
spm_pobs(
  x = NULL,
  y = NULL,
  aH = -0.05,
  aL = -0.01,
  f1H = 60,
  f1L = 80,
  QH = 2e-08,
  QL = 2.5e-08,
  fH = 60,
  fL = 80,
  bH = 4,
  bL = 5,
  mu0H = 8e-06,
  mu0L = 1e-05,
  thetaH = 0.08,
  thetaL = 0.1,
  p = 0.25,
  stopifbound = FALSE,
  algorithm = "NLOPT_LN_NELDERMEAD",
  lb = NULL,
  ub = NULL,
  maxeval = 500,
  verbose = FALSE,
  pinv.tol = 0.01,
  mode = "observed",
  gomp = TRUE,
  ftol_rel = 1e-06
)
```
Arguments

- **x**: A data table with genetic component.
- **y**: A data table without genetic component.
- **aH**: A k by k matrix. Characterizes the rate of the adaptive response for \( Z = 1 \).
- **aL**: A k by k matrix. Characterizes the rate of the adaptive response for \( Z = 0 \).
- **f1H**: A deviation from the norm (or optimal) state for \( Z = 1 \). This is a vector of length \( k \).
- **f1L**: A deviation from the norm (or optimal) state for \( Z = 0 \). This is a vector of length \( k \).
- **QH**: A matrix k by k, which is a non-negative-definite symmetric matrix for \( Z = 1 \).
- **QL**: A matrix k by k, which is a non-negative-definite symmetric matrix for \( Z = 0 \).
- **fH**: A vector with length of \( k \). Represents the normal (or optimal) state for \( Z = 1 \).
- **fL**: A vector with length of \( k \). Represents the normal (or optimal) state for \( Z = 0 \).
- **bH**: A diffusion coefficient, k by k matrix for \( Z = 1 \).
- **bL**: A diffusion coefficient, k by k matrix for \( Z = 0 \).
- **mu0H**: A baseline mortality for \( Z = 1 \).
- **mu0L**: A baseline mortality for \( Z = 0 \).
- **thetaH**: A displacement coefficient for \( Z = 1 \).
- **thetaL**: A displacement coefficient for \( Z = 0 \).
- **p**: A hypothetical percentage of presence of partially observed covariate in a population (default \( p = 0.25 \)).
- **stopifbound**: If TRUE then estimation stops if at least one parameter achieves lower or upper boundaries.
- **algorithm**: An optimization algorithm used, can be one of those provided by nloptr. #Check the NLopt website for a description of the algorithms. Default: NLOPT_LN_NELDERMEAD
- **lb**: Lower bound of parameter values.
- **ub**: Upper bound of parameter values.
- **maxeval**: Maximum number of iterations of the algorithm for nloptr optimization. The program stops when the number of function evaluations exceeds maxeval. Default: 500.
- **verbose**: An indicator of verbosing output (FALSE by default).
- **mode**: Can be one of the following: "observed" (default), "unobserved" or "combined". mode = "observed" represents analysing only dataset with observed variable \( Z \). mode = "unobserved" represents analysing only dataset of unobserved variable \( Z \). mode = "combined" denoted joint analysis of both observed and unobserved datasets.
- **gomp**: A flag (FALSE by default). When it is set, then time-dependent exponential form of \( mu0 \) is used: \( mu0 = mu0*exp(theta*t) \).
- **ftol_rel**: Relative tolerance threshold for likelihood function (defalult: 1e-6), see http://ab-initio.mit.edu/wiki/index.php/NLopt_Reference
Value

A set of estimated parameters $a_H$, $a_L$, $f_{1H}$, $f_{1L}$, $Q_H$, $Q_L$, $f_H$, $f_L$, $b_H$, $b_L$, $\mu_0H$, $\mu_0L$, $\theta_H$, $\theta_L$, $p$ and additional variable $\text{limit}$ which indicates if any parameter achieved lower or upper boundary conditions (FALSE by default).

References


Examples

```r
## Not run:
library(stpm)
#Reading the data:
data <- sim_pobs(N=1000)
head(data)
#Parameters estimation:
pars <- spm_pobs(x=data)
pars

## End(Not run)
```

spm_projection

A data projection with previously estimated or user-defined parameters. Projections are constructed for a cohort with fixed or normally distributed initial covariates.

Description

A data projection with previously estimated or user-defined parameters. Projections are constructed for a cohort with fixed or normally distributed initial covariates.

Usage

```r
spm_projection(
  x,
  N = 100,
  ystart = 80,
  model = "discrete",
  tstart = 30,
  tend = 105,
  dt = 1,
  sd0 = 1,
  nobs = NULL,
  gomp = TRUE,
  format = "short"
)
```
**Arguments**

- **x**
  A list of parameters from output of the `spm(...)` function.

- **N**
  A number of individuals to simulate, N=100 by default.

- **ystart**
  A vector of starting values of covariates (variables), ystart=80 by default.

- **model**
  A model type. Choices are: "discrete", "continuous" or "time-dependent".

- **tstart**
  Start time (age), default=30. Can be an interval: c(a, b) - in this case, the starting time is simulated via `runif(1, a, b)`.

- **tend**
  End time (age), default=105.

- **dt**
  A time interval between observations, dt=1 by default.

- **sd0**
  A standard deviation value for simulation of the next value of variable. sd0=1 by default.

- **nobs**
  A number of observations (lines) for i-th individual.

- **gomp**
  A flag (FALSE by default). When it is set, then time-dependent exponential form of mu0 and Q are used: mu0 = mu0*exp(theta*t), Q = Q*exp(theta*t). Only for continuous-time SPM.

- **format**
  Data format: "short" (default), "long".

**Value**

An object of `spm.projection` class with two elements. (1) A simulated data set. (2) A summary statistics which includes (i) age-specific means of state variables and (ii) Survival probabilities.

**References**


**Examples**

```r
## Not run:
library(stpm)
set.seed(123)
# Setting up the model
model.par <- list()
model.par$a <- matrix(c(-0.05, 1e-3, 2e-3, -0.05), nrow=2, ncol=2, byrow=TRUE)
model.par$f1 <- matrix(c(90, 35), nrow=1, ncol=2)
model.par$Q <- matrix(c(1e-8, 1e-9, 1e-9, 1e-8), nrow=2, ncol=2, byrow=TRUE)
model.par$f <- matrix(c(80, 27), nrow=1, ncol=2)
model.par$b <- matrix(c(0, 2), nrow=2, ncol=2)
model.par$mu0 <- 1e-6
model.par$theta <- 0.09
```
# Projection
# Discrete-time model
data.proj.discrete <- spm_projection(model.par, N=5000, ystart=c(80, 27))
plot(data.proj.discrete$stat$srv.prob)

# Continuous-time model
data.proj.continuous <- spm_projection(model.par, N=5000, ystart=c(80, 27), model="continuous")
plot(data.proj.continuous$stat$srv.prob)

# Time-dependent model
model.par <- list(at = "-0.05", f1t = "80", Qt = "2e-8",
ft= "80", bt = "5", mu0t = "1e-5*exp(0.11*t)")
data.proj.time_dependent <- spm_projection(model.par, N=500, ystart=80, model="time-dependent")
plot(data.proj.time_dependent$stat$srv.prob, xlim = c(30,105))

## End(Not run)

**spm_time_dep**

A function for the model with time-dependent model parameters.

## Description

A function for the model with time-dependent model parameters.

## Usage

```r
spm_time_dep(
  x,
  start = list(a = -0.05, f1 = 80, Q = 2e-08, f = 80, b = 5, mu0 = 0.001),
  frm = list(at = "a", f1t = "f1", Qt = "Q", ft = "f", bt = "b", mu0t = "mu0"),
  stopifbound = FALSE,
  lb = NULL,
  ub = NULL,
  verbose = FALSE,
  opts = list(algorithm = "NLOPT_LN_NELDERMEAD", maxeval = 100, ftol_rel = 1e-08),
  lrtest = FALSE
)
```

## Arguments

- **x**: Input data table.
- **start**: A list of starting parameters, default: `start=list(a=-0.5, f1=80, Q=2e-8, f=80, b=5, mu0=1e-5)`.
- **frm**: A list of formulas that define age (time) dependency. Default: `frm=list(at="a", f1t="f1", Qt="Q", ft="f", bt="b", mu0t="mu0")`.
- **stopifbound**: Estimation stops if at least one parameter achieves lower or upper boundaries. Default: `FALSE`.
1b  Lower bound of parameters under estimation.
ub  Upper bound of parameters under estimation.
verbose  Turns on verbosing output.
opts  A list of options for nloptr. Default value: opt=list(algorithm="NLOPT_LN_NELDERMEAD", maxeval=100, ftol_rel=1e-8).
lrtest  Indicates should Likelihood-Ratio test be performed. Possible values: TRUE, H01, H02, H03, H04, H05 (see package Vignette for details) Default value: FALSE. Please see nloptr documentation for more information.

Value

A set of estimates of a, f1, Q, f, b, mu0.
status  Optimization status (see documentation for nloptr package).
LogLik  A logarithm likelihood.
objective  A value of objective function (given by nloptr).
message  A message given by nloptr optimization function (see documentation for nloptr package).

References


Examples

library(stpm)
set.seed(123)
#Data preparation:
n <- 5
data <- simdata_time_dep(N=n)
# Estimation:
opt.par <- spm_time_dep(data)
opt.par

---

**stpm**  
**Stochastic Process Model for Analysis of Longitudinal and Time-to-Event Outcomes**

**Description**

Utilities to estimate parameters of the models with survival functions induced by stochastic covariates. Miscellaneous functions for data preparation and simulation are also provided. For more information, see: "Stochastic model for analysis of longitudinal data on aging and mortality" by Yashin A. et al, 2007, Mathematical Biosciences, 208(2), 538-551 <DOI:10.1016/j.mbs.2006.11.006>.

**Author(s)**

References


Examples

```r
## Not run:
library(stpm)
#Prepare data for optimization
data <- prepare_data(x=system.file("extdata","longdat.csv",package="stpm"), covariates="BMI")
#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("extdata","longdat.csv",package="stpm"), covariates="BMI")
p.td.model <- spm(data, model="time-dependent")
p.td.model
## End(Not run)
```

---

`trim`  
Returns string w/o leading or trailing whitespace

### Description

Returns string w/o leading or trailing whitespace

### Usage

`trim(x)`

### Arguments

`x` a string to trim
trim.leading  

**Description**

Returns string w/o leading whitespace

**Usage**

trim.leading(x)

**Arguments**

x  
a string to trim

trim.trailing  

**Description**

Returns string w/o trailing whitespace

**Usage**

trim.trailing(x)

**Arguments**

x  
a string to trim

vitstat  

**Vital (mortality) statistics.**

**Description**

Vital (mortality) statistics.

**Author(s)**

Ilya Y Zhbannikov <ilya.zhbannikov@duke.edu>
Index

* censoring
  stpm, 40
* data
  ex_data, 3
  longdat, 7
  vitstat, 42
* longitudinal
  stpm, 40
* modeling
  stpm, 40
* stochastic
  stpm, 40
* time-to-event
  stpm, 40

ex_data, 3
func1, 3
get.column.index, 4
getNextY.cont, 4
getNextY.cont2, 5
getNextY.disc, 5
getNextY.disc.m, 6
getPrevY.disc, 6
getPrevY.disc.m, 7
longdat, 7
LRTest, 8
m, 8
make.short.format, 9
mu, 9

prepare_data, 10
prepare_data_cont, 11
prepare_data_discr, 12

sigma_sq, 13
sim_pobs, 19
simdata_cont, 13

simdata_discr, 15
simdata_gamma_frailty, 16
simdata_time_dep, 17
spm, 21
spm.impute, 23
spm_con_1d, 30
spm_con_1d_g, 32
spm_cont_lin, 26
spm_cont_quad_lin, 28
spm_continuous, 24
spm_discrete, 34
spm_pobs, 35
spm_projection, 37
spm_time_dep, 39
stpm, 40
stpm-package (stpm), 40

trim, 41
trim.leading, 42
trim.trailing, 42
vitstat, 42