Package ‘lmenssp’

July 24, 2016

Type Package

Title Linear Mixed Effects Models with Non-Stationary Stochastic Processes

Version 1.2

Date 2016-07-23

Author Ozgur Asar, Peter J. Diggle

Maintainer Ozgur Asar <ozgur.asar@acibadem.edu.tr>

Depends MASS, nlme, mvtnorm, geoR,

Description Contains functions to estimate model parameters and filter, smooth and forecast random effects coefficients for mixed models with stationary and non-stationary stochastic processes under multivariate normal and t response distributions, diagnostic checks, bootstrap standard error calculation, etc.

NeedsCompilation no

Repository CRAN

Date/Publication 2016-07-24 00:16:30

R topics documented:

lmenssp-package	*Linear Mixed Effects Models with Non-stationary Stochastic Processes*

Description

Obtains maximum likelihood estimates of the model parameters, filters, smooths and forecasts random components of the model for the following processes: 1) Brownian motion, 2) integrated Brownian motion, 3) integrated Ornstein-Uhlenbeck process, 4) stationary process with powered correlation function, 5) stationary process with Matern correlation function, under multivariate normal and t response distributions. It also contains miscellaneous functions for diagnostic checks, boostrap standard error calculation, etc.

Details

Package:	lmenssp
Type:	Package
Version:	1.2
Date:	2016-07-23
License:	GPL (>=2)

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Diggle PJ (1988) An approach to the analysis of repeated measurements. *Biometrics*, **44**, 959-971.

Diggle PJ, Heagerty PJ, Liang K-Y, Zeger SL. (2002) *Analysis of Longitudinal Data, 2nd edition*. Oxford University Press: Oxford.

Diggle PJ, Ribeiro PJ Jr. (2007) *Model-based Geostatistics*. Springer-Verlag: New York.

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Laird NM, Ware JH (1982) Random-effects models for longitudinal data. *Biometrics*, **38**, 134-147.

Matern B. (1960) *Spatial Variation*. Statens Skogsforsningsinstitut, Stockholm.

Pinheiro JC, Liu C, Wu YN. (2001) Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution. *Journal of Computational and Graphical Statistics* **10**, 249-276.

Ross SM (1996) *Stochastic processes*. John Wiley & Sons, New Jersey.

Taylor JMG, Cumberland WG, Sy JP (1994) A Stochastic Model for Analysis of Longitudinal AIDS Data. *Journal of the American Statistical Association*, **89**, 727-736.

boot.nm *A function to calculate bootstrap standard errors*

Description

Calculates bootstrap standard errors for the parameter estimates obtained by `lmenssp` when Nelder-Mead algorithm is used

Usage

```
boot.nm(formula, data, id, timeVar, result, matern = TRUE, kappa.or.power,
nboot = 100, tol.lmenssp = 1e-08, maxiter.lmenssp = 500)
```

Arguments

`formula`	a typical `R` formula for the fixed effects component of the model
`data`	a data frame from which the variables are to be extracted
`id`	a numerical vector for subject identification
`timeVar`	a numerical vector for the time variable
`result`	a matrix of results obtained by `lmenssp`, see the example below
`matern`	a logical variable, `TRUE` corresponds to Matern correlation function, `FALSE` corresponds to powered correlation function
`kappa.or.power`	a numerical value for the shape parameter, it corresponds to κ if `matern = TRUE` and ϕ if `matern = FALSE`
`nboot`	a numerical value for number of bootstrap sample
`tol.lmenssp`	a numerical value for the tolerance, to be passed to `lmenssp`
`maxiter.lmenssp`	a numerical value for the maximum number of iterations, to be passed to `lmenssp`

Details

This function consider parametric bootstrap based on the fitted model. The recommended number of bootstrap replications is at least 100. For the details of κ and ϕ in `kappa.or.power`, see the details section of `lmenssp` function.

Value

Returns a list of results

Author(s)

Ozgur Asar, Peter J. Diggle

Examples

```
# loading the data set and subsetting it for the first 5 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm)
data.sim.ibm.short <- data.sim.ibm[data.sim.ibm$id <= 5, ]

# model formula to be used below
formula <- log.egfr ~ sex + bage + fu + pwl

# fitting the mixed model with Matern, kappa = 0.5
fit.matern <- lmenssp(formula = formula, data = data.sim.ibm.short,
  id = data.sim.ibm.short$id, process = "sgp-matern-0.5", timeVar = data.sim.ibm.short$fu,
  init = c(-13, 1, -1), silent = FALSE)
fit.matern

# bootstrapping the standard errors, nboot is set to 2 for illustration
# set nboot to at least 100 in your applications
fit.matern.boot <- boot.nm(formula = formula, data = data.sim.ibm.short,
                           id = data.sim.ibm.short$id, timeVar = data.sim.ibm.short$fu,
                           result = fit.matern$est, matern = TRUE, kappa.or.power = 0.5,
                           nboot = 2)
fit.matern.boot
```

`data.sim.ibm` *A simulated data set under a mixed model with random intercept, integrated Brownian motion and multivariate normal response distribution*

Description

A longitudinal data set containing measurements on 500 subjects, each with their own sequence of follow-up time points collected at irregularly spaced follow-up time points.

Usage

```
data("data.sim.ibm")
```

Format

A data frame with 8462 observations on the following 6 variables.

`id` a numeric vector: subject identification

`sex` a numeric vector: 0 = male, 1 = female

`bage` a numeric vector: baseline age in years

`fu` a numeric vector: follow-up time in years

`pwl` a numeric vector: piecewise linear term, calculated as, max(0, age-56.5)

`log.egfr` a numeric vector: response variable representing natural logarithm of estimated glomerular filtration rate

Details

For the details of the true parameter settings, see Diggle, Sousa and Asar (2015).

References

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Examples

```
data(data.sim.ibm)
str(data.sim.ibm)
```

---

data.sim.ibm.heavy — *A simulated data set under a mixed model with random intercept, integrated Brownian motion and multivariate t response distribution*

Description

A longitudinal data set containing measurements on 100 subjects, each with their own sequence of follow-up time points collected at irregularly spaced follow-up time points.

Usage

```
data("data.sim.ibm.heavy")
```

Format

A data frame with 1636 observations on the following 6 variables.

- `id` a numeric vector: subject identification
- `sex` a numeric vector: 0 = male, 1 = female
- `bage` a numeric vector: baseline age in years
- `fu` a numeric vector: follow-up time in years
- `pwl` a numeric vector: piecewise linear term, calculated as, max(0, age-56.5)
- `log.egfr` a numeric vector: response variable representing natural logarithm of estimated glomerular filtration rate

Details

For the true parameter settings of the fixed effects and variance parameters, see Diggle, Sousa and Asar (2015). True degrees-of-freedom parameter is set to 5.

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Examples

```
data(data.sim.ibm.heavy)
str(data.sim.ibm.heavy)
```

`filtered` *A function for filtering under multivariate normal response distribution*

Description

Filters random components of the mixed model with a stationary or non-stationary stochastic process component, under multivariate normal response distribution

Usage

```
filtered(formula, data = NULL, id, process = "bm", timeVar, estimate, subj.id)
```

Arguments

`formula`	a typical `R` formula for the fixed effects component of the model
`data`	a data frame from which the variables are to be extracted
`id`	a vector for subject identification
`process`	a character string, `"bm"` for Brownian motion, `"ibm"` for integrated Brownian motion, `"iou"` for integrated Ornstein-Uhlenbeck process, `"sgp-powered-power-method"` for stationary process with powered correlation function, and `"sgp-matern-kappa"` for stationary process with Matern correlation function
`timeVar`	a vector for the time variable at which filtered values are required
`estimate`	a vector for the maximum likelihood estimates
`subj.id`	a vector for IDs of the subjects for whom filtering will be carried out

Details

For the details of `process`, see `lmenssp`.

Value

Returns the results as lists for the random intercept and stochastic process

Author(s)

Ozgur Asar, Peter J. Diggle

References

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm)
data.sim.ibm.short <- data.sim.ibm[data.sim.ibm$id <= 20, ]

# model formula to be used below
formula <- log.egfr ~ sex + bage + fu + pwl

# obtaining the maximum likelihood estimates of the model
# parameters for the model with integrated Brownian motion
fit.ibm <- lmenssp(formula = formula, data = data.sim.ibm.short,
  id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu,
  silent = FALSE)
fit.ibm

# filtering for subjects with ID=1 and 2
subj.id <- c(1, 2)
fil.res <- filtered(formula = formula, data = data.sim.ibm.short,
    id = data.sim.ibm.short$id, process = "ibm",  timeVar = data.sim.ibm.short$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = subj.id)
fil.res

# filtering for a new (hypothetical) subject
data.501   <- data.frame(id = c(501, 501, 501), sex = c(0, 0, 0),
   bage = c(50, 50, 50), fu = c(0, 0.2, 0.4),
   pwl = c(0, 0, 0), log.egfr = c(4.3, 2.1, 4.1))
data.501
fil.501 <- filtered(formula = formula, data = data.501,
   id = data.501$id, process = "ibm", timeVar = data.501$fu,
   estimate = fit.ibm$estimate[, 1], subj.id = 501)
fil.501
```

---

lmenssp — *Function to obtain the maximum likelihood estimates of the parameters for linear mixed effects models with random intercept and a stationary or non-stationary stochastic process component, under multivariate normal response distribution*

Description

Obtains the maximum likelihood estimates of the parameters for linear mixed effects models with random intercept and a stationary or non-stationary stochastic process component, under multivariate normal response distribution

Usage

```
lmenssp(formula, data = NULL, id, process = "bm", timeVar, init = NULL, tol = 1e-05,
        maxiter = 100, silent = TRUE)
```

Arguments

`formula`	a typical `R` formula for the fixed effects component of the model
`data`	a data frame from which the variables are to be extracted
`id`	a numerical vector for subject identification
`process`	a character string for the stochastic process: `"bm"` for Brownian motion, `"ibm"` for integrated Brownian motion, `"iou"` for integrated Ornstein-Uhlenbeck process, `"sgp-powered-power-method"` for stationary process with powered correlation function (for the usage, see the details below) and `"sgp-matern-kappa"` for stationary process with Matern correlation function (for the usage, see the details below)
`timeVar`	a numerical vector for the time variable
`init`	a numerical vector of initial values for the variance parameters to start the Fisher-Scoring or Nelder-Mead algorithms; if the user does not provide their own values `lmenssp` obtains these by fitting a random intercept and random slope model using the `lme` function of the **nlme** package; for details see the details below
`tol`	a numerical value for the maximum tolerance to assess the convergence
`maxiter`	a numerical value for the number of iterations for the Fisher-Scoring or Nelder-Mead algorithms
`silent`	a character string, if set to `FALSE` the details of the Fisher-Scoring steps are printed when the algorithm is running

Details

For `"process"`, `"sgp-powered-power-method"` is a general form for stationary process with powered correlation function. `"power"` is the shapre parameter and corresponds to `"c"` in

$$exp(-|t-s|^c/\phi),$$

where t and s are two time points and ϕ is the scale parameter, and `"method"` might be `"fs"` for Fisher-Scoring or `"nm"` for Nelder-Mead. Some examples are: `"sgp-powered-1-fs"` for stationary process with exponential correlation function with Fisher-Scoring algorithm and `"sgp-powered-2-nm"` for stationary process with Gaussian correlation function with Nelder-Mead algorithm. Similarly, `"sgp-matern-kappa"` is a general form for stationary process with Matern correlation function. `"kappa"` is the shape parameter and corresponds to κ in

$$\left\{2^{\kappa-1}\Gamma\left(\kappa\right)\right\}^{-1}\left(|t-s|/\nu\right)^{\kappa}K_{\kappa}\left(|t-s|/\nu\right),$$

t and s are two time points and ν is the scale parameter. An example is "sgp-matern-0.5" for stationary process with exponential correlation function. Nelder-Mead algorithm is automatically specified for the choice of Matern, i.e. Fisher-Scoring is not available.

"init" assumes the following:

- 3 element vectors for "process = bm" and "process = ibm", with initials for the variances of random intercept, stochastic process and measurement error, respectively

- 4 element vector for "process = iou", with initials for the variance of random intercept variance, (two) parameters of the stochastic process, variance of measurement error

- 3 element vector for "process = sgp-powered-power-fs", with initials for log of the fraction of the variance of random intercept and variance of the process, log(ϕ) and log of the fraction of the variance of measurement error and variance of the process.

- NULL for any of the specification of "process", in which case lmenssp finds the initals internally using the lme function of the **nlme** package.

Value

Returns the results as lists

Author(s)

Ozgur Asar, Peter J. Diggle

References

Diggle PJ (1988) An approach to the analysis of repeated measurements. *Biometrics*, **44**, 959-971.

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Taylor JMG, Cumberland WG, Sy JP (1994) A Stochastic Model for Analysis of Longitudinal AIDS Data. *Journal of the American Statistical Association*, **89**, 727-736.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm)
data.sim.ibm.short <- data.sim.ibm[data.sim.ibm$id <= 20, ]

# fitting the model with integrated Brownian motion
fit.ibm <- lmenssp(log.egfr ~ sex + bage + fu + pwl, data = data.sim.ibm.short,
  id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu,
  silent = FALSE)
fit.ibm
```

lmenssp.heavy *Function to obtain the maximum likelihood estimates of the parameters for linear mixed effects models with random intercept and a stationary or non-stationary stochastic process component, under multivariate t response distribution*

Description

Obtains the maximum likelihood estimates of the parameters by expectatio-maximisation (E-M) algorithm for linear mixed effects models with random intercept and a stationary or non-stationary stochastic process component, under multivariate normal t distribution

Usage

```
lmenssp.heavy(formula, data, id, timeVar, init.em = NULL, maxiter.em = 100,
tol.em = 0.001, process, silent = TRUE, dof.est = c(0.1, 10, 1e-04),
tol.cd = 0.001, tol.lmenssp = 10^-5, init.lmenssp = NULL, maxiter.lmenssp = 100,
silent.lmenssp = TRUE)
```

Arguments

`formula`	a typical `R` formula for the fixed effects component of the model
`data`	a data frame from which the variables are to be extracted
`id`	a numerical vector for subject identification
`timeVar`	a numerical vector for the time variable
`init.em`	a vector of inital values for the E-M algorithm
`maxiter.em`	a numerical value for the maximum number of iterations for the E-M algorithm
`tol.em`	a numerical value for the maximum tolerance to assess the convergence of the E-M algorithm
`process`	a character string for the stochastic process: `"bm"` for Brownian motion, `"ibm"` for integrated Brownian motion, `"iou"` for integrated Ornstein-Uhlenbeck process, `"sgp-powered-power-method"` for stationary process with powered correlation function and `"sgp-matern-kappa"` for stationary process with Matern correlation function
`silent`	a character string, if set to `FALSE` the details of the E-M steps are printed when the algorithm is running
`dof.est`	a vector of three elements, to be passed to `optimize`
`tol.cd`	a numerical value for the tolerance of central-difference approximation
`tol.lmenssp`	a numerical value for the maximum tolerance to assess the convergence, to be passed to `lmenssp`
`init.lmenssp`	a vector of inital values, to be passed to `lmenssp`
`maxiter.lmenssp`	a numerical value of the number of iterations for the Fisher-Scoring or Nelder-Mead algorithms, to be passed to `lmenssp`

silent.lmenssp — a character string, if set to FALSE the details of the Fisher-Scoring steps are printed when the algorithm is running, to be passed to lmenssp

Details

lmenssp.heavy calls lmenssp inside.

Whilst theoretical standard errors are calculated and reported only for the fixed effects, central-difference based standard errors are calculated and reported for all the parameter estimates.

There are more than one way of specifying init.em, it can be set to:

1) fixed effects, random effects parameters and degrees-of freedom,

2) only the degrees-of-freedom, and

3) NULL; for this specification, lmenssp.heavy finds the intials internally.

For the details of "process", see lmenssp.

In dof.est, first and second elements are the minimum and maximum values of the search and the third element is the tolerance. It is passed to optimize.

Value

Returns a list of results.

Author(s)

Ozgur Asar, Peter J. Diggle

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Pinheiro JC, Liu C, Wu YN. (2001) Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution. *Journal of Computational and Graphical Statistics* **10**, 249-276.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm.heavy)
data.sim.ibm.heavy.short <- data.sim.ibm.heavy[data.sim.ibm.heavy$id <= 20, ]

# estimating the parameters
# tol.em is set to 10^-1 and tol.lmenssp to 10^-2 only for illustration,
# decrease these values in your applications
fit.heavy <- lmenssp.heavy(formula = log.egfr ~ sex + bage + fu + pwl,
       data = data.sim.ibm.heavy.short, id = data.sim.ibm.heavy.short$id,
       timeVar = data.sim.ibm.heavy.short$fu, init.em = 5, maxiter.em = 1000,
       tol.em = 10^-1, process = "ibm", silent = FALSE,
       dof.est = c(0.1, 10, 0.0001), tol.cd = 0.001, tol.lmenssp = 10^-2,
       silent.lmenssp = FALSE)
fit.heavy
```

`qqplot.t` *Quantile-quantile plot for univariate t distribution*

Description

Calculates empirical quantiles of univariate data and theoretical quantiles of a t distribution with a given degrees-of-freedom

Usage

```
qqplot.t(x, dof, print = FALSE)
```

Arguments

`x`	a vector or column matrix for data
`dof`	a numerical value for degrees-of-freedom
`print`	a logical variable, if "`print = TRUE`" empirical and theoretical quantiles are reported and the user can use them to draw a new plot

Value

a list of output is returned if "`print = TRUE`"

Author(s)

Ozgur Asar, Peter J. Diggle

See Also

`qqplot`

Examples

```
set.seed(1)
rs <- rt(500, 5)
qqplot.t(rs, 5)
```

`smoothed` *A function for smoothing under multivariate normal response distribution*

Description

Smooths random components of the mixed model with a stationary or non-stationary stochastic process component, under multivariate normal response distribution

Usage

```
smoothed(formula, data = NULL, id, process = "bm", timeVar, estimate,
subj.id = NULL, fine = NULL, eq.forec = NULL, uneq.forec = NULL)
```

Arguments

`formula`	a typical R formula for the fixed effects component of the model
`data`	a data frame from which the variables are to be extracted
`id`	a vector for subject identification
`process`	a character string, `"bm"` for Brownian motion, `"ibm"` for integrated Brownian motion, `"iou"` for integrated Ornstein-Uhlenbeck process, `"sgp-powered-power-method"` for stationary process with powered correlation function, and `"sgp-matern-kappa"` for stationary process with Matern correlation function
`timeVar`	a vector for the time variable
`estimate`	a vector for the maximum likelihood estimates
`fine`	a numerical value for smoothing at fine intervals within the follow-up period
`subj.id`	a vector of IDs of the subject for whom smoothing is to be carried out
`eq.forec`	a two element vector for equally spaced forecasting
`uneq.forec`	a two-column data frame or matrix for forecasting at desired time points

Details

For details of `"process"` see `lmenssp`.

Value

Returns the results as lists for the random intercept and stochastic process

Author(s)

Ozgur Asar, Peter J. Diggle

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Diggle PJ (1988) An approach to the analysis of repeated measurements. *Biometrics*, **44**, 959-971.

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm)
data.sim.ibm.short <- data.sim.ibm[data.sim.ibm$id <= 20, ]

# model formula to be used below
formula <- log.egfr ~ sex + bage + fu + pwl

# obtaining the maximum likelihood estimates of the model
# parameters for the model with integrated Brownian motion
fit.ibm <- lmenssp(formula = formula, data = data.sim.ibm.short,
  id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu, silent = FALSE)
fit.ibm

# smoothing for subject with ID=1 and 2
subj.id <- c(1, 2)
smo.res <- smoothed(formula = formula, data = data.sim.ibm.short,
    id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = subj.id)
smo.res

# smoothing with fine interval of 0.01 within the follow-up period
smo.within <- smoothed(formula = formula, data = data.sim.ibm.short,
    id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = subj.id, fine = 0.01)
smo.within

# one, two and three month forecasting for patients with IDs = 1 and 2
eq.forecast <- smoothed(formula = formula, data = data.sim.ibm.short,
    id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = subj.id,
    eq.forec = c(1/12, 3))
eq.forecast

# forecasting at arbitrary time points for patients with IDs = 1 and 2
uneq.forec <- data.frame(c(1, 1, 1, 2, 2), c(1/12, 2/12, 6/12, 1/12, 3/12))
uneq.forecast <- smoothed(formula = formula, data = data.sim.ibm.short,
    id = data.sim.ibm.short$id, process = "ibm", timeVar = data.sim.ibm.short$fu,
    estimate = fit.ibm$estimate[, 1], uneq.forec = uneq.forec)
uneq.forecast

## smoothing for a new (hypothetical) patient
```

```
data.501   <- data.frame(id = c(501, 501, 501), sex = c(0, 0, 0),
  bage = c(50, 50, 50), fu = c(0, 0.2, 0.4),
  pwl = c(0, 0, 0), log.egfr = c(4.3, 2.1, 4.1))
new.id     <-  501

# at observed time points
smo.501 <- smoothed(formula = formula, data = data.501,
    id = data.501$id, process = "ibm", timeVar = data.501$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = new.id)
smo.501

# at fine interval of 0.01 within the follow-up period
smo.within.501 <- smoothed(formula = formula, data = data.501,
    id = data.501$id, process = "ibm", timeVar = data.501$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = new.id, fine = 0.01)
smo.within.501

# one, two and three month forecasting
eq.forecast.501 <- smoothed(formula = formula, data = data.501,
    id = data.501$id, process = "ibm", timeVar = data.501$fu,
    estimate = fit.ibm$estimate[, 1], subj.id = new.id,
    eq.forec = c(1/12, 3))
eq.forecast.501

# forecasting at arbitrary time points
uneq.forec.501 <- data.frame(c(501, 501, 501), c(1/12, 2/12, 4/12))
uneq.forecast.501 <- smoothed(formula = formula, data = data.501,
    id = data.501$id, process = "ibm", timeVar = data.501$fu,
    estimate = fit.ibm$estimate[, 1], uneq.forec = uneq.forec.501)
uneq.forecast.501
```

`smoothed.heavy` *A function for smoothing under multivariate t response distribution*

Description

Smooths random components of the mixed model with a stationary or non-stationary stochastic process component, under multivariate t response distribution

Usage

```
smoothed.heavy(formula, data, id, process, timeVar, estimate, subj.id = NULL)
```

Arguments

`formula`	a typical R formula for the fixed effects component of the model
`data`	a data frame from which the variables are to be extracted
`id`	a vector for subject identification

process	a character string, "bm" for Brownian motion, "ibm" for integrated Brownian motion, "iou" for integrated Ornstein-Uhlenbeck process, "sgp-powered-power-method" for stationary process with powered correlation function, and "sgp-matern-kappa" for stationary process with Matern correlation function
timeVar	a vector for the time variable
estimate	a vector for the maximum likelihood estimates
subj.id	a vector of IDs of the subject for whom smoothing is to be carried out

Details

For details of "process" see lmenssp

Value

Returns the results as lists for the random intercept and stochastic process

Author(s)

Ozgur Asar, Peter J. Diggle

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Pinheiro JC, Liu C, Wu YN. (2001) Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution. *Journal of Computational and Graphical Statistics* **10**, 249-276.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm.heavy)
data.sim.ibm.heavy.short <- data.sim.ibm.heavy[data.sim.ibm.heavy$id <= 20, ]

# a formula to be used
formula <- log.egfr ~ sex + bage + fu + pwl

# estimating the parameters
# tol.em is set to 10^-1 and tol.lmenssp to 10^-2 only for illustration,
# decrease these values in your applications
fit.heavy <- lmenssp.heavy(formula = formula, data = data.sim.ibm.heavy.short,
      id = data.sim.ibm.heavy.short$id, timeVar = data.sim.ibm.heavy.short$fu, init.em = 5,
       maxiter.em = 1000, tol.em = 10^-1,
       process = "ibm", silent = FALSE, dof.est = c(0.1, 10, 0.0001), tol.cd = 0.001,
       tol.lmenssp = 10^-2, silent.lmenssp = FALSE)
fit.heavy

# smoothing for the patients with ID = 1, 2, 3, 4
smo.heavy <- smoothed.heavy(formula = formula, data = data.sim.ibm.heavy.short,
```

```
    id = data.sim.ibm.heavy.short$id, process = "ibm", timeVar = data.sim.ibm.heavy.short$fu,
     estimate = fit.heavy$est, subj.id = c(1, 2, 3, 4))
smo.heavy
```

`var.inspect` *A function for calculating empirical variances with respect to time for data sets with regularly or irregularly spaced follow-up time points*

Description

Calculates empirical variances for data sets with regularly or irregularly spaced time points, and plots the result

Usage

```
var.inspect(resid, timeVar, binwidth, numElems = 0, irregular = T)
```

Arguments

`resid`	a vector of empirical residuals
`timeVar`	a vector for the time variable
`binwidth`	a numerical value for the bin length, to be used for irregularly spaced data
`numElems`	a numerical value for the elimination of the bins with less than that number of elements
`irregular`	a character string, `FALSE` indicates the data are collected at regular time points

Value

Returns mid values and variances of the bins, and numbers of elements falling into the bins for `irregular = TRUE`, and unique time points and variances, and number of the elements for the time points for `irregular = FALSE`.

Author(s)

Ozgur Asar, Peter J. Diggle

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm)
data.sim.ibm.short <- data.sim.ibm[data.sim.ibm$id <= 20, ]

# obtaining empirical residuals by a linear model
# and calculating the empirical variances
lm.fit <- lm(log.egfr ~ sex + bage + fu + pwl, data = data.sim.ibm.short)
var.inspect(resid = resid(lm.fit), timeVar = data.sim.ibm.short$fu, binwidth = 0.1,
   numElems = 20, irregular = TRUE)
```

---

variogram — *A function for calculating the empirical variogram for data sets with regularly or irregularly spaced follow-up time points*

Description

Calculates empirical variogram for data sets with regularly or irregularly spaced time points, and plots the result

Usage

```
variogram(resid, timeVar, id, binwidth, numElems = 0, inc.var = TRUE,
irregular = TRUE)
```

Arguments

`resid`	a vector of empirical residuals
`timeVar`	a vector for the time variable
`id`	a vector for subject identification
`binwidth`	a numerical value for the bin length, to be used for irregularly spaced data
`numElems`	a numerical value for the elimination of the bins with less than that number of elements
`inc.var`	a character string, if set `FALSE` process variance is not included in the plot
`irregular`	a character string, `FALSE` indicates the data are collected at regular time points

Value

Returns mid values and means of the bins, and numbers of elements falling into the bins for `irregular = TRUE`, and lags and means of the lags, and number of the elements for the lags for `irregular = FALSE`. Process variance is also returned.

Author(s)

Ozgur Asar, Peter J. Diggle

References

Asar O, Ritchie J, Kalra P, Diggle PJ (2015) Acute kidney injury amongst chronic kidney disease patients: a case-study in statistical modelling. To be submitted.

Diggle PJ (1988) An approach to the analysis of repeated measurements. *Biometrics*, **44**, 959-971.

Diggle PJ, Heagerty PJ, Liang K-Y, Zeger SL. (2002) *Analysis of Longitudinal Data, 2nd edition*. Oxford University Press: Oxford.

Diggle PJ, Ribeiro PJ Jr. (2007) *Model-based Geostatistics*. Springer-Verlag: New York.

Diggle PJ, Sousa I, Asar O (2015) Real time monitoring of progression towards renal failure in primary care patients. *Biostatistics*, 16(3), 522-536.

Examples

```
# loading the data set and subsetting it for the first 20 patients
# for the sake illustration of the usage of the functions
data(data.sim.ibm)
data.sim.ibm.short <- data.sim.ibm[data.sim.ibm$id <= 20, ]

# obtaining empirical residuals by a linear model
# and calculating the empirical variogram
lm.fit <- lm(log.egfr ~ sex + bage + fu + pwl, data = data.sim.ibm.short)
variogram(resid = resid(lm.fit), timeVar = data.sim.ibm.short$fu, id = data.sim.ibm.short$id,
  binwidth = 0.1, numElems = 20, inc.var = FALSE)
```

Index