Package 'lqmm'

November 7, 2021

Type Package

Title Linear Quantile Mixed Models

Version 1.5.6

Date 2021-11-04

Author Marco Geraci

Maintainer Marco Geraci <marco.geraci@uniroma1.it>

Depends R (>= 3.0.0)

Imports stats, utils, nlme (>= 3.1-124), SparseGrid

Description Functions to fit quantile regression models for hierarchical data (2-level nested designs) as described in Geraci and Bottai (2014, Statistics and Computing) <doi:10.1007/s11222-013-9381-9>. A vignette is given in Geraci (2014, Journal of Statistical Software) <doi:10.18637/jss.v057.i13> and included in the package documents. The packages also provides functions to fit quantile models for independent data and for count responses.

License GPL (>= 2)

LazyLoad yes

RoxygenNote 7.0.2

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-11-07 16:00:06 UTC

R topics documented:

lqmm-package 2
boot 3
coef.lqm 5
coef.lqmm 6
covHandling 6
dal 7

extractBoot . . . 8
gauss.quad . . . 9
gauss.quad.prob . . . 9
is.positive.definite . . . 10
labor . . . 10
logLik.lqm . . . 11
logLik.lqmm . . . 12
lqm . . . 12
lqm.counts . . . 14
lqm.fit.gs . . . 16
lqmControl . . . 18
lqmm . . . 19
lqmm.fit.df . . . 23
lqmm.fit.gs . . . 25
lqmmControl . . . 26
make.positive.definite . . . 28
meanAL . . . 28
mleAL . . . 29
Orthodont . . . 30
predict.lqm . . . 31
predict.lqmm . . . 32
print.lqm . . . 33
print.lqmm . . . 34
print.summary.lqm . . . 35
print.summary.lqmm . . . 35
ranef.lqmm . . . 36
residuals.lqm . . . 37
residuals.lqmm . . . 37
summary.boot.lqm . . . 38
summary.boot.lqmm . . . 39
summary.lqm . . . 40
summary.lqmm . . . 41
VarCorr.lqmm . . . 42

Index **44**

`lqmm-package` *Linear Quantile Models and Linear Quantile Mixed Models*

Description

Fit quantile regression models for independent and hierarchical data

Details

Package:	lqmm
Type:	Package
Version:	1.5.6
Date:	2021-11-04
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

Marco Geraci

Maintainer: Marco Geraci <geraci@mailbox.sc.edu>

References

Geraci M (2014). Linear quantile mixed models: The lqmm package for Laplace quantile regression. Journal of Statistical Software, 57(13), 1–29. <doi:10.18637/jss.v057.i13>

Geraci M and Bottai M (2007). Quantile regression for longitudinal data using the asymmetric Laplace distribution. Biostatistics 8(1), 140–154. <doi:10.1093/biostatistics/kxj039>

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479. <doi:10.1007/s11222-013-9381-9>.

`boot` *Bootstrap functions for LQM and LQMM*

Description

This function is used to obtain a bootstrap sample of a fitted LQM or LQMM. It is a generic function.

Usage

```
boot(object, R = 50, seed = round(runif(1, 1, 10000)), startQR = FALSE)
## S3 method for class 'lqm'
boot(object, R = 50, seed = round(runif(1, 1, 10000)), startQR = FALSE)
## S3 method for class 'lqmm'
boot(object, R = 50, seed = round(runif(1, 1, 10000)), startQR = FALSE)
```

Arguments

object	an object of class "lqm" or "lqmm".
R	number of bootstrap replications.
seed	optional random number generator seed.
startQR	logical flag. If TRUE the estimated parameters in object are used as starting values in the algorithm applied to each bootstrap sample. This may cause the algorithm to converge too often to a similar optimum, which would ultimately result in underestimated standard errors. If FALSE (recommended), starting values are based on lm.

Value

An object of class boot.lqm is a data frame with R rows and npars columns containing the bootstrap estimates of theta. If object contains results for multiple quantiles, boot.lqm returns an array of dimension c(R,npars,nt), where nt is the length of tau.

An object of class boot.lqmm is a data frame with R rows and npars columns containing the bootstrap estimates of theta_x, theta_z, and scale. If object contains results for multiple quantiles, boot.lqmm returns an array of dimension c(R,npars,nt), where nt is the length of tau. The elements of theta_z are labelled with reStruct. See function covHandling and the example below on how to derive the variance-covariance matrix of the random effects starting from theta_z.

The following attributes are available:

tau	index of the quantile(s).
estimated	the estimated parameter as given by object.
R	number of bootstrap replications.
seed	the random number generator seed used to produce the bootstrap sample.
npars	total numer of parameters.
rdf	the number of residual degrees of freedom.
indices	the bootstrap sample of independent data units.

Author(s)

Marco Geraci

Examples

```
# boot.lqm
set.seed(123)
n <- 500
test <- data.frame(x = runif(n,0,1))
test$y <- 30 + test$x + rnorm(n)
fit.lqm <- lqm(y ~ x, data = test, tau = 0.5)
fit.boot <- boot(fit.lqm)
str(fit.boot)
```

```
# boot.lqmm
data(Orthodont)
fit <- lqmm(distance ~ age, random = ~ 1, group = Subject,
tau = 0.5, data = Orthodont)
fit.boot <- boot(fit)
str(fit.boot)
```

`coef.lqm` *Extract LQM Coefficients*

Description

`coef` extracts model coefficients from `lqm`, `lqm.counts` objects.

Usage

```
## S3 method for class 'lqm'
coef(object, ...)
```

Arguments

`object`	an `lqm` or `lqm.counts` object.
`...`	not used.

Value

a vector for single quantiles or a matrix for multiple quantiles.

Author(s)

Marco Geraci

See Also

`lqm` `summary.lqm` `lqm.counts`

coef.lqmm *Extract LQMM Coefficients*

Description

coef extracts model coefficients from lqmm objects.

Usage

```
## S3 method for class 'lqmm'
coef(object, ...)
```

Arguments

object	a fitted object of class "lqmm".
...	not used.

Value

a vector for single quantiles or a matrix for multiple quantiles.

Author(s)

Marco Geraci

See Also

lqmm summary.lqmm

covHandling *Variance-Covariance Matrix*

Description

This is an auxiliary function.

Usage

```
covHandling(theta, n, cov_name, quad_type)
```

Arguments

theta	unique parameters of the variance-covariance matrix of the random effects as returned by lqmm in theta_z.
n	dimension of the vector of random effects.
cov_name	see argument covariance in lqmm.
quad_type	type of quadrature "c("normal","robust")".

Author(s)

Marco Geraci

See Also

VarCorr.lqmm

---

dal *The Asymmetric Laplace Distribution*

Description

Density, distribution function, quantile function and random generation for the asymmetric Laplace distribution.

Usage

```
dal(x, mu = 0, sigma = 1, tau = 0.5, log = FALSE)
pal(x, mu = 0, sigma = 1, tau = 0.5)
qal(x, mu = 0, sigma = 1, tau = 0.5)
ral(n, mu = 0, sigma = 1, tau = 0.5)
```

Arguments

x	vector of quantiles (dal, pal) or probabilities (qal).
n	number of observations.
mu	location parameter.
sigma	positive scale parameter.
tau	skewness parameter (0,1).
log	logical; if TRUE, probabilities are log–transformed.

Details

The asymmetric Laplace distribution with parameters (mu, sigma, tau) has density

$$f(x) = \tau(1-\tau)/\sigma e^{-1/(2\sigma)(\theta max(x,0)+(1-\theta)max(-x,0))}$$

Author(s)

Marco Geraci

See Also

lqmm, lqm

extractBoot *Extract Fixed and Random Bootstrapped Parameters*

Description

This generic function extracts the fixed and random components of bootstrapped estimates of an lqmm object.

Usage

```
extractBoot(object, which = "fixed")
## S3 method for class 'boot.lqmm'
extractBoot(object, which = "fixed")
```

Arguments

object	an object of class boot.lqmm.
which	character indicating whether "fixed" or "random" parameters.

Details

The "random" parameters refer to the "raw" parameters of the variance-covariance matrix of the random effects as returned by lqmm.fit.gs and lqmm.fit.df.

Value

a matrix of bootstrapped estimates.

Author(s)

Marco Geraci

See Also

boot.lqmm, lqmm.fit.gs, lqmm.fit.df

Examples

```
## Orthodont data
data(Orthodont)

# Random intercept model
fit <- lqmm(distance ~ age, random = ~ 1, group = Subject,
tau = 0.5, data = Orthodont)
fit.boot <- boot(fit)

# extract fixed effects
B <- extractBoot(fit.boot, which = "fixed")
```

```
# covariance matrix estimated fixed parameters
cov(B)
```

---

gauss.quad *Gaussian Quadrature*

Description

This function calculates nodes and weights for Gaussian quadrature. See `help("gauss.quad")` from package `statmod`.

Author(s)

Original version by Gordon Smyth

Source

Gordon Smyth with contributions from Yifang Hu, Peter Dunn and Belinda Phipson. (2011). statmod: Statistical Modeling. R package version 1.4.11. https://CRAN.R-project.org/package=statmod

---

gauss.quad.prob *Gaussian Quadrature*

Description

This function calculates nodes and weights for Gaussian quadrature in terms of probability distributions. See `help("gauss.quad.prob")` from package `statmod`.

Author(s)

Original version by Gordon Smyth

Source

Gordon Smyth with contributions from Yifang Hu, Peter Dunn and Belinda Phipson. (2011). statmod: Statistical Modeling. R package version 1.4.11. https://CRAN.R-project.org/package=statmod

is.positive.definite *Test for Positive Definiteness*

Description

This function tests whether all eigenvalues of a symmetric matrix are positive. See `help("is.positive.definite")` from package `corpcor`.

Author(s)

Original version by Korbinian Strimmer

Source

Juliane Schaefer, Rainer Opgen-Rhein, Verena Zuber, A. Pedro Duarte Silva and Korbinian Strimmer. (2011). corpcor: Efficient Estimation of Covariance and (Partial) Correlation. R package version 1.6.0. https://CRAN.R-project.org/package=corpcor

labor *Labor Pain Data*

Description

The `labor` data frame has 358 rows and 4 columns of the change in pain over time for several 83 women in labor.

Format

This data frame contains the following columns:

subject an ordered factor indicating the subject on which the measurement was made. The levels are labelled `1` to `83`.

pain a numeric vector of self–reported pain scores on a 100mm line.

treatment a dummy variable with values `1` for subjects who received a pain medication and `0` for subjects who received a placebo.

time a numeric vector of times (minutes since randomization) at which `pain` was measured.

Details

The labor pain data were reported by Davis (1991) and successively analyzed by Jung (1996) and Geraci and Bottai (2007). The data set consists of repeated measurements of self–reported amount of pain on N = 83 women in labor, of which 43 were randomly assigned to a pain medication group and 40 to a placebo group. The response was measured every 30 min on a 100–mm line, where 0 means no pain and 100 means extreme pain. A nearly monotone pattern of missing data was found for the response variable and the maximum number of measurements for each woman was six.

Source

Davis CS (1991). Semi–parametric and non–parametric methods for the analysis of repeated measurements with applications to clinical trials. Statistics in Medicine 10, 1959–80.

References

Geraci M and Bottai M (2007). Quantile regression for longitudinal data using the asymmetric Laplace distribution. Biostatistics 8(1), 140–154.

Jung S (1996). Quasi–likelihood for median regression models. Journal of the American Statistical Association 91, 251–7.

`logLik.lqm` *Extract Log-Likelihood*

Description

`logLik.lqm` extracts the log-likelihood of a fitted LQM.

Usage

```
## S3 method for class 'lqm'
logLik(object, ...)
```

Arguments

`object`	an object of `class` "lqm".
`...`	not used.

Author(s)

Marco Geraci

See Also

`lqm` `AIC`

logLik.lqmm *Extract Log-Likelihood*

Description

logLik.lqmm extracts the log-likelihood of a fitted LQMM.

Usage

```
## S3 method for class 'lqmm'
logLik(object, ...)
```

Arguments

object	an object of class "lqmm".
...	not used.

Author(s)

Marco Geraci

See Also

lqmm AIC

lqm *Fitting Linear Quantile Models*

Description

lqm is used to fit linear quantile models based on the asymmetric Laplace distribution.

Usage

```
lqm(formula, data, subset, na.action, weights = NULL, tau = 0.5,
contrasts = NULL, control = list(), fit = TRUE)
```

Arguments

formula	an object of class formula for fixed effects: a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lqm is called.

subset	an optional vector specifying a subset of observations to be used in the fitting process.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options.
weights	An optional vector of weights to be used in the fitting process.
tau	the quantile(s) to be estimated. This must be a number between 0 and 1, otherwise the execution is stopped. If more than one quantile is specified, rounding off to the 4th decimal must give non–duplicated values of tau, otherwise the execution is stopped.
contrasts	an optional list. See the contrasts.arg of model.matrix.default.
control	list of control parameters of the fitting process. See lqmControl.
fit	logical flag. If FALSE the function returns a list of arguments to be passed to lqm.fit.gs.

Details

The function computes an estimate on the tau-th quantile function of the response, conditional on the covariates, as specified by the formula argument. The quantile predictor is assumed to be linear. The function maximizes the (log)likelihood of a Laplace regression which is equivalent to the minimization of the weighted sum of absolute residuals (Koenker and Bassett, 1978). The optimization algorithm is based on the gradient of the Laplace log–likelihood (Bottai, Orsini and Geraci, 2013).

Value

lqm returns an object of class lqm.

The function summary is used to obtain and print a summary of the results.

An object of class lqm is a list containing the following components:

theta	a vector of coefficients. theta is a named matrix of coefficients when tau is a vector of values.
scale	the scale parameter.
gradient	the gradient.
logLik	the log–likelihood.
opt	details on optimization (see lqm.fit.gs).
call	the matched call.
term.labels	names for theta.
terms	the terms object used.
nobs	the number of observations.
edf,dim_theta	the length of theta.
rdf	the number of residual degrees of freedom.
tau	the estimated quantile(s).
x	the model matrix.

y	the model response.
weights	the weights used in the fitting process (a vector of 1's if weights = NULL).
InitialPar	starting values for theta.
control	list of control parameters used for optimization (see lqmControl).

Note

Updates/FAQ/news are published here https://marcogeraci.wordpress.com/. New versions are usually published here https://github.com/marco-geraci/lqmm/ before going on CRAN.

Author(s)

Marco Geraci

References

Bottai M, Orsini N, Geraci M (2015). A Gradient Search Maximization Algorithm for the Asymmetric Laplace Likelihood, Journal of Statistical Computation and Simulation, 85(10), 1919-1925.

Chen C (2007). A finite smoothing algorithm for quantile regression. Journal of Computational and Graphical Statistics, 16(1), 136-164.

Koenker R and Bassett G (1978). Regression Quantiles. Econometrica 46(1), 33–50.

See Also

summary.lqm, coef.lqm, predict.lqm, residuals.lqm

Examples

```
set.seed(123)
n <- 500
p <- 1:3/4
test <- data.frame(x = runif(n,0,1))
test$y <- 30 + test$x + rnorm(n)
fit.lqm <- lqm(y ~ x, data = test, tau = p,
control = list(verbose = FALSE, loop_tol_ll = 1e-9), fit = TRUE)
fit.lqm
```

---

lqm.counts *Quantile Regression for Counts*

Description

This function is used to fit a quantile regression model when the response is a count variable.

Usage

```
lqm.counts(formula, data, weights = NULL, offset = NULL, contrasts = NULL,
tau = 0.5, M = 50, zeta = 1e-05, B = 0.999, cn = NULL, alpha = 0.05,
control = list())
```

Arguments

formula	an object of class formula: a symbolic description of the model to be fitted.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lqm is called.
weights	an optional vector of weights to be used in the fitting process.
offset	an optional offset to be included in the model frame.
contrasts	an optional list. See the contrasts.arg of model.matrix.default.
tau	quantile to be estimated.
M	number of dithered samples.
zeta	small constant (see References).
B	right boundary for uniform random noise U[0,B] to be added to the response variable (see References).
cn	small constant to be passed to F.lqm (see References).
alpha	significance level.
control	list of control parameters of the fitting process. See lqmControl.

Details

A linear quantile regression model if fitted to the log–transformed response. Additional tranformation functions will be implemented. The notation used here follows closely that of Machado and Santos Silva (2005).

Value

an object of class "lqm.counts" containing the following components

tau	the estimated quantile.
theta	regression quantile (on the log–scale).
fitted	predicted quantile (on the response scale).
tTable	coefficients, standard errors, etc.
x	the model matrix.
y	the model response.
offset	offset.
nobs	the number of observations.
M	specified number of dithered samples for standard error estimation.

Mn	actual number of dithered samples used for standard error estimation that gave an invertible D matrix (Machado and Santos Silva, 2005).
term.labels	names for theta.
terms	the terms object used.
rdf	the number of residual degrees of freedom.
InitialPar	starting values for theta.
control	list of control parameters used for optimization (see lqmControl).

Author(s)

Marco Geraci

References

Machado JAF and Santos Silva JMC (2005). Quantiles for counts. Journal of the American Statistical Association, 100(472), 1226–1237.

Examples

```
n <- 100
x <- runif(n)
test <- data.frame(x = x, y = rpois(n, 2*x))
lqm.counts(y ~ x, data = test, M = 50)
```

---

lqm.fit.gs *Quantile Regression Fitting by Gradient Search*

Description

This function controls the arguments to be passed to routines written in C for LQM estimation. The optimization algorithm is based on the gradient of the Laplace log–likelihood (Bottai, Orsini and Geraci, 2013).

Usage

```
lqm.fit.gs(theta, x, y, weights, tau, control)
```

Arguments

theta	starting values for the regression coefficients.
x	the model matrix.
y	the model response.
weights	the weights used in the fitting process.
tau	the quantile to be estimated.
control	list of control parameters used for optimization (see lqmControl).

Details

See argument `fit` in `lqm` for generating a list of arguments to be called by this function.

Value

An object of class `list` containing the following components:

`theta`	a vector of coefficients.
`scale`	the scale parameter.
`gradient`	the gradient.
`logLik`	the log–likelihood.
`opt`	number of iterations when the estimation algorithm stopped.

.

Author(s)

Marco Geraci

References

Bottai M, Orsini N, Geraci M (2014). A Gradient Search Maximization Algorithm for the Asymmetric Laplace Likelihood, Journal of Statistical Computation and Simulation, 85, 1919-1925.

See Also

`lqm`

Examples

```
set.seed(123)
n <- 500
test <- data.frame(x = runif(n,0,1))
test$y <- 30 + test$x + rnorm(n)
lqm.ls <- lqm(y ~ x, data = test, fit = FALSE)

do.call("lqm.fit.gs", lqm.ls)
```

`lqmControl` *Control parameters for lqm estimation*

Description

A list of parameters for controlling the fitting process.

Usage

```
lqmControl(method = "gs1", loop_tol_ll = 1e-5, loop_tol_theta = 1e-3,
check_theta = FALSE, loop_step = NULL, beta = 0.5, gamma = 1.25,
reset_step = FALSE, loop_max_iter = 1000, smooth = FALSE,
omicron = 0.001, verbose = FALSE)
```

Arguments

`method`	character vector that specifies which code to use for carrying out the gradient search algorithm: "gs1" (default) based on C code and "gs2" based on R code. Method "gs3" uses a smoothed loss function. See details.
`loop_tol_ll`	tolerance expressed as relative change of the log-likelihood.
`loop_tol_theta`	tolerance expressed as relative change of the estimates.
`check_theta`	logical flag. If `TRUE` the algorithm performs a check on the change in the estimates in addition to the likelihood.
`loop_step`	step size (default standard deviation of response).
`beta`	decreasing step factor for line search (0,1).
`gamma`	nondecreasing step factor for line search (>= 1).
`reset_step`	logical flag. If `TRUE` the step size is re-setted to the initial value at each iteration.
`loop_max_iter`	maximum number of iterations.
`smooth`	logical flag. If `TRUE` the standard loss function is replaced with a smooth approximation.
`omicron`	small constant for smoothing the loss function when using `smooth = TRUE`. See details.
`verbose`	logical flag.

Details

The methods "gs1" and "gs2" implement the same algorithm (Bottai et al, 2015). The former is based on C code, the latter on R code. While the C code is faster, the R code seems to be more efficient in handling large datasets. For method "gs2", it is possible to replace the classical non-differentiable loss function with a smooth version (Chen, 2007).

Value

a list of control parameters.

Author(s)

Marco Geraci

References

Bottai M, Orsini N, Geraci M (2015). A Gradient Search Maximization Algorithm for the Asymmetric Laplace Likelihood, Journal of Statistical Computation and Simulation, 85(10), 1919-1925.

Chen C (2007). A finite smoothing algorithm for quantile regression. Journal of Computational and Graphical Statistics, 16(1), 136-164.

See Also

lqm

---

lqmm *Fitting Linear Quantile Mixed Models*

Description

lqmm is used to fit linear quantile mixed models based on the asymmetric Laplace distribution.

Usage

```
lqmm(fixed, random, group, covariance = "pdDiag", tau = 0.5,
nK = 7, type = "normal", rule = 1, data = sys.frame(sys.parent()),
subset, weights, na.action = na.fail, control = list(),
contrasts = NULL, fit = TRUE)
```

Arguments

fixed	an object of class formula for fixed effects: a symbolic description of the model to be fitted.
random	a one-sided formula of the form ~x1 + x2 + ... + xn for random effects: a symbolic description of the model to be fitted.
group	grouping factor.
covariance	variance–covariance matrix of the random effects. Default is pdDiag (see details).
tau	the quantile(s) to be estimated.
nK	number of quadrature knots.
type	type of quadrature "c("normal","robust")" (see details).
rule	quadrature rule (see details).
data	an optional data frame containing the variables named in fixed, random and group. By default the variables are taken from the environment from which lqmm is called.

subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process of the same length as the number of rows of data. Weights are given to clusters, therefore units within the same cluster receive the same weight (see details).
na.action	a function that indicates what should happen when the data contain NAs. The default action (na.fail) causes lqmm to print an error message and terminate if there are any incomplete observations.
control	list of control parameters of the fitting process. See lqmmControl.
contrasts	not yet implemented.
fit	logical flag. If FALSE the function returns a list of arguments to be passed to lqmm.fit.

Details

The function computes an estimate on the tau-th quantile function of the response, conditional on the covariates, as specified by the formula argument, and on random effects, as specified by the random argument. The quantile predictor is assumed to be linear. The function maximizes the (log)likelihood of the Laplace regression proposed by Geraci and Bottai (2014). The likelihood is numerically integrated via Gaussian quadrature techniques. The optimization algorithm is based on the gradient of the Laplace log–likelihood (control = list(method = "gs")). An alternative optimization algorithm is based on a Nelder-Mead algorithm (control = list(method = "df")) via optim. The scale parameter is optimized in a refinement step via optimize.

Quadrature approaches include Gauss-Hermite (type = "normal") and Gauss-Laguerre (type = "robust") quadrature. The argument rule takes one of the following: 1 (product rule quadrature), 2 (sparse grid quadrature), 3 (nested quadrature rule - only for type = "normal"), 4 (quadrature rule with the smallest number of nodes between rules 1 or 2). Rules 2 and 3 have not yet been tested extensively.

Different standard types of positive–definite matrices for the random effects can be specified: pdIdent multiple of an identity; pdCompSymm compound symmetry structure (constant diagonal and constant off–diagonal elements); pdDiag diagonal; pdSymm general positive–definite matrix, with no additional structure.

Weights are given to clusters, therefore it is expected that these are constant within cluster. When the weights are specified in the main call, then the first value by group in the vector weights will be replicated for the same length of each group. Alternatively, different weights within the same cluster can be introduced with a direct call to lqmm.fit.gs or lqmm.fit.df.

The lqmm vignette can be accessed by typing help(package = "lqmm") and then following the link 'User guides, package vignettes and other documentation'.

Value

lqmm returns an object of class lqmm.

The function summary is used to obtain and print a summary of the results.

An object of class lqmm is a list containing the following components:

theta	a vector containing fixed regression coefficients and parameters of the variance–covariance matrix of the random effects. See VarCorr.lqmm to extract the variance–covariance of the random effects from an "lqmm" object.
theta_x,theta_z	partition of theta: fixed regression coefficients (theta_x) and unique variance–covariance parameters (theta_z).
scale	the scale parameter.
gradient	the gradient (control = list(method = "gs")).
logLik	the log–likelihood.
opt	details on optimization (see lqmm.fit.gs and lqmm.fit.df).
call	the matched call.
nn	column names of mmf.
mm	column names of mmr.
nobs	the number of observations.
dim_theta	the number of columns in mmf and mmr.
dim_theta_z	the length of theta_z.
edf	length of theta.
rdf	the number of residual degrees of freedom.
df	edf + 1 (scale parameter).
tau	the estimated quantile(s).
mmf	the model matrix – fixed effects.
mmr	the model matrix – random effects.
y	the model response.
revOrder	original order of observations (now ordered according to group).
weights	the likelihood weights used in the fitting process (a vector of 1's if weights is missing or NULL).
group	the grouping factor.
ngroups	the number of groups.
QUAD	quadrature nodes and weights.
type	the type of quadrature.
rule	quadrature rule.
InitialPar	starting values for theta.
control	list of control parameters used for optimization (see lqmmControl).
cov_name	class of variance-covariance matrix for the random effects.
mfArgs	arguments for model.frame to return the full data frame.

Note

Updates/FAQ/news are published here https://marcogeraci.wordpress.com/. New versions are usually published here https://github.com/marco-geraci/lqmm/ before going on CRAN.

Author(s)

Marco Geraci

References

Genz A, and Keister BD (1996). Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight. Journal of Computational and Applied Mathematics, 71(2), 299–309. <doi:10.1016/0377-0427(95)00232-4>

Geraci M (2014). Linear quantile mixed models: The lqmm package for Laplace quantile regression. Journal of Statistical Software, 57(13), 1–29. <doi:10.18637/jss.v057.i13>

Geraci M and Bottai M (2007). Quantile regression for longitudinal data using the asymmetric Laplace distribution. Biostatistics 8(1), 140–154. <doi:10.1093/biostatistics/kxj039>

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479. <doi:10.1007/s11222-013-9381-9>.

Heiss F, and Winschel V (2008). Likelihood approximation by numerical integration on sparse grids. Journal of Econometrics, 144(1), 62–80. <doi:10.1016/j.jeconom.2007.12.004>

See Also

lqm, summary.lqmm, coef.lqmm, VarCorr.lqmm, predict.lqmm, residuals.lqmm

Examples

```
# Test example
set.seed(123)

M <- 50
n <- 10
test <- data.frame(x = runif(n*M,0,1), group = rep(1:M,each=n))
test$y <- 10*test$x + rep(rnorm(M, 0, 2), each = n) + rchisq(n*M, 3)
fit.lqmm <- lqmm(fixed = y ~ x, random = ~ 1, group = group,
data = test, tau = 0.5, nK = 11, type = "normal")
fit.lqmm

#Call: lqmm(fixed = y ~ x, random = ~1, group = group, tau = 0.5, nK = 11,
#    type = "normal", data = test)
#Quantile 0.5

#Fixed effects:
#(Intercept)           x
#      3.443       9.258

#Covariance matrix of the random effects:
#(Intercept)
#      3.426

#Residual scale parameter: 0.8697 (standard deviation 2.46)
#Log-likelihood: -1178
```

```
#Number of observations: 500
#Number of groups: 50


## Orthodont data
data(Orthodont)

# Random intercept model
fitOi.lqmm <- lqmm(distance ~ age, random = ~ 1, group = Subject,
tau = c(0.1,0.5,0.9), data = Orthodont)
coef(fitOi.lqmm)

# Random slope model
fitOs.lqmm <- lqmm(distance ~ age, random = ~ age, group = Subject,
tau = c(0.1,0.5,0.9), cov = "pdDiag", data = Orthodont)

# Extract estimates
VarCorr(fitOs.lqmm)
coef(fitOs.lqmm)
ranef(fitOs.lqmm)

# AIC
AIC(fitOi.lqmm)
AIC(fitOs.lqmm)
```

`lqmm.fit.df` *Linear Quantile Mixed Models Fitting by Derivative-Free Optimization*

Description

This function controls the arguments to be passed to `optim` and `optimize` for LQMM estimation.

Usage

```
lqmm.fit.df(theta_0, x, y, z, weights, cov_name, V, W, sigma_0,
tau, group, control)
```

Arguments

`theta_0`	starting values for the linear predictor.
`x`	the model matrix for fixed effects (see details).
`y`	the model response (see details).
`z`	the model matrix for random effects (see details).
`weights`	the weights used in the fitting process (see details).
`cov_name`	variance–covariance matrix of the random effects. Default is `pdIdent`. See details.

V	nodes of the quadrature.
W	weights of the quadrature.
sigma_0	starting value for the scale parameter.
tau	the quantile(s) to be estimated.
group	the grouping factor (see details).
control	list of control parameters used for optimization (see lqmmControl).

Details

In lqmm, see argument fit for generating a list of arguments to be called by this function; see argument covariance for alternative variance–covariance matrices.

NOTE: the data should be ordered by group when passed to lqmm.fit.df (such ordering is performed by lqmm).

Value

An object of class "list" containing the following components:

theta	a vector of coefficients, including the "raw" variance–covariance parameters (see VarCorr.lqmm).
scale	the scale parameter.
logLik	the log–likelihood.
opt	number of iterations when the estimation algorithm stopped for lower (theta) and upper (scale) loop.

.

Author(s)

Marco Geraci

See Also

lqmm

Examples

```
set.seed(123)

M <- 50
n <- 10
test <- data.frame(x = runif(n*M,0,1), group = rep(1:M,each=n))
test$y <- 10*test$x + rep(rnorm(M, 0, 2), each = n) + rchisq(n*M, 3)
lqmm.ls <- lqmm(fixed = y ~ x, random = ~ 1, group = group, data = test,
fit = FALSE)

do.call("lqmm.fit.df", lqmm.ls)
```

lqmm.fit.gs *Linear Quantile Mixed Models Fitting by Gradient Search*

Description

This function controls the arguments to be passed to routines written in C for LQMM estimation. The optimization algorithm is based on the gradient of the Laplace log–likelihood (Bottai, Orsini and Geraci, 2014; Geraci and Bottai, 2014).

Usage

```
lqmm.fit.gs(theta_0, x, y, z, weights, cov_name, V, W, sigma_0, tau,
group, control)
```

Arguments

theta_0	starting values for the linear predictor.
x	the model matrix for fixed effects (see details).
y	the model response (see details).
z	the model matrix for random effects (see details).
weights	the weights used in the fitting process (see details).
cov_name	variance–covariance matrix of the random effects. Default is pdIdent. See details.
V	nodes of the quadrature.
W	weights of the quadrature.
sigma_0	starting value for the scale parameter.
tau	the quantile(s) to be estimated.
group	the grouping factor (see details).
control	list of control parameters used for optimization (see lqmmControl).

Details

In lqmm, see argument fit for generating a list of arguments to be called by this function; see argument covariance for alternative variance–covariance matrices.

NOTE: the data should be ordered by group when passed to lqmm.fit.gs (such ordering is performed by lqmm).

Value

An object of class "list" containing the following components:

theta	a vector of coefficients, including the "raw" variance–covariance parameters (see VarCorr.lqmm).

scale	the scale parameter.
gradient	the gradient.
logLik	the log–likelihood.
opt	number of iterations when the estimation algorithm stopped for lower (theta) and upper (scale) loop.

.

Author(s)

Marco Geraci

References

Bottai M, Orsini N, Geraci M. (2014). A gradient search maximization algorithm for the asymmetric Laplace likelihood, Journal of Statistical Computation and Simulation (in press).

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479.

See Also

lqmm

Examples

```
set.seed(123)

M <- 50
n <- 10
test <- data.frame(x = runif(n*M,0,1), group = rep(1:M,each=n))
test$y <- 10*test$x + rep(rnorm(M, 0, 2), each = n) + rchisq(n*M, 3)
lqmm.ls <- lqmm(fixed = y ~ x, random = ~ 1, group = group,
data = test, fit = FALSE)

do.call("lqmm.fit.gs", lqmm.ls)
```

---

lqmmControl *Control parameters for lqmm estimation*

Description

A list of parameters for controlling the fitting process.

Usage

```
lqmmControl(method = "gs", LP_tol_ll = 1e-5, LP_tol_theta = 1e-5,
check_theta = FALSE, LP_step = NULL, beta = 0.5, gamma = 1,
reset_step = FALSE, LP_max_iter = 500, UP_tol = 1e-4,
UP_max_iter = 20, startQR = FALSE, verbose = FALSE)
```

Arguments

method	character vector that specifies the estimation method: "gs" for gradient search (default) and "df" for Nelder-Mead.
LP_tol_ll	tolerance expressed as absolute change of the log-likelihood.
LP_tol_theta	tolerance expressed as absolute change of theta
check_theta	logical flag. If TRUE the algorithm performs an additional check on the change in the estimates.
LP_step	step size (default standard deviation of response).
beta	decreasing step factor for line search (0,1).
gamma	nondecreasing step factor for line search (>= 1).
reset_step	logical flag. If TRUE the step size is reset to the initial value at each iteration.
LP_max_iter	maximum number of iterations
UP_tol	tolerance expressed as absolute change of the scale parameter.
UP_max_iter	maximum number of iterations.
startQR	logical flag. If FALSE (default) the least squares estimate of the fixed effects is used as starting value of theta_x and scale. If TRUE the lqm estimate is used.
verbose	logical flag.

Details

LP (lower loop) refers to the estimation of regression coefficients and variance-covariance parameters. UP (upper loop) refers to the estimation of the scale parameter.

Value

a list of control parameters.

Author(s)

Marco Geraci

See Also

lqmm

make.positive.definite *Compute Nearest Positive Definite Matrix*

Description

This function computes the nearest positive definite of a real symmetric matrix. See help("make.positive.definite") from package corpcor.

Author(s)

Original version by Korbinian Strimmer

Source

Juliane Schaefer, Rainer Opgen-Rhein, Verena Zuber, A. Pedro Duarte Silva and Korbinian Strimmer. (2011). corpcor: Efficient Estimation of Covariance and (Partial) Correlation. R package version 1.6.0. https://CRAN.R-project.org/package=corpcor

meanAL *Functions for Asymmetric Laplace Distribution Parameters*

Description

Accessory functions.

Usage

```
meanAL(mu, sigma, tau)
varAL(sigma, tau)
invvarAL(x, tau)
```

Arguments

mu	location parameter.
sigma	scale parameter.
tau	skewness parameter.
x	numeric value.

Details

meanAL computes the mean of an asymmetric Laplace with parameters mu, sigma and tau.

varAL computes the variance of an asymmetric Laplace with parameters sigma and tau.

invvarAL computes the scale parameter of an asymmetric Laplace with parameter tau and variance x.

Author(s)

Marco Geraci

References

Yu K and Zhang J (2005). A three-parameter asymmetric Laplace distribution and its extension. Communications in Statistics-Theory and Methods 34, 1867–1879.

See Also

dal, mleAL

---

mleAL *Maximum Likelihood Estimation of Asymmetric Laplace Distribution*

Description

This function estimates the parameters of an asymmetric Laplace distribution for a sample.

Usage

```
mleAL(x)
```

Arguments

x	a numeric vector.

Value

an object of class `list` containing the following components:

m	location parameter
sigma	scale parameter
tau	skewness parameter
r	number of iterations

Author(s)

Marco Geraci

References

Yu K and Zhang J (2005). A three-parameter asymmetric Laplace distribution and its extension. Communications in Statistics-Theory and Methods 34, 1867–1879.

See Also

dal, meanAL

`Orthodont` *Growth curve data on an orthdontic measurement*

Description

The `Orthodont` data frame has 108 rows and 4 columns of the change in an orthdontic measurement over time for several young subjects.

Format

This data frame contains the following columns:

distance a numeric vector of distances from the pituitary to the pterygomaxillary fissure (mm). These distances are measured on x-ray images of the skull.

age a numeric vector of ages of the subject (yr).

Subject an ordered factor indicating the subject on which the measurement was made. The levels are labelled `M01` to `M16` for the males and `F01` to `F13` for the females. The ordering is by increasing average distance within sex.

Sex a factor with levels `Male` and `Female`

Details

Investigators at the University of North Carolina Dental School followed the growth of 27 children (16 males, 11 females) from age 8 until age 14. Every two years they measured the distance between the pituitary and the pterygomaxillary fissure, two points that are easily identified on x-ray exposures of the side of the head.

Source

Pinheiro, J. C. and Bates, D. M. (2000), *Mixed-Effects Models in S and S-PLUS*, Springer, New York. (Appendix A.17)

Potthoff, R. F. and Roy, S. N. (1964), "A generalized multivariate analysis of variance model useful especially for growth curve problems", Biometrika, 51, 313–326.

Jose Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar and the R Development Core Team (2011). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-100. https://CRAN.R-project.org/package=nlme

predict.lqm *Predictions from LQM Objects*

Description

This function computes predictions based on fitted linear quantile model.

Usage

```
## S3 method for class 'lqm'
predict(object, newdata, interval = FALSE,
level = 0.95, na.action = na.pass, ...)
## S3 method for class 'lqm.counts'
predict(object, newdata,
na.action = na.pass, ...)
```

Arguments

object	an lqm or lqm.counts object.
newdata	an optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
interval	logical flag. If TRUE, bootstrap percentile intervals for predictions are provided. This argument is for lqm objects only.
level	confidence level. This argument is for lqm objects only.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
...	further arguments passed to boot.lqm.

Value

a vector or a matrix or an array of predictions.

Author(s)

Marco Geraci

See Also

residuals.lqm, residuals.lqm.counts, lqm, lqm.counts, coef.lqm, boot.lqm

predict.lqmm *Predictions from an* lqmm *Object*

Description

The predictions at level 0 correspond to predictions based only on the fixed effects estimates. The predictions at level 1 are obtained by adding the best linear predictions of the random effects to the predictions at level 0. See details for interpretation. The function predint will produce 1-alpha confidence intervals based on bootstrap centiles.

Usage

```
## S3 method for class 'lqmm'
predict(object, newdata, level = 0,
na.action = na.pass, ...)
## S3 method for class 'lqmm'
predint(object, level = 0, alpha = 0.05,
R = 50, seed = round(runif(1, 1, 10000)))
```

Arguments

object	an lqmm object.
newdata	an optional data frame in which to look for variables with which to predict. If omitted, the fitted values are produced.
level	an optional integer vector giving the level of grouping to be used in obtaining the predictions.
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
alpha	1-alpha is the confidence level.
R	number of bootstrap replications.
seed	optional random number generator seed.
...	not used.

Details

As discussed by Geraci and Bottai (2014), integrating over the random effects will give "weighted averages" of the cluster-specific quantile effects. These may be interpreted strictly as population regression quantiles for the median (tau=0.5) only. Therefore, predictions at the population level (code=0) should be interpreted analogously.

Value

a vector or a matrix of predictions for predict.lqmm. A data frame or a list of data frames for predint.lqmm containing predictions, lower and upper bounds of prediction intervals, and standard errors.

Author(s)

Marco Geraci

References

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479.

See Also

lqmm, ranef.lqmm, coef.lqmm

Examples

```
## Orthodont data
data(Orthodont)

# Random intercept model
fitOi.lqmm <- lqmm(distance ~ age, random = ~ 1, group = Subject,
tau = c(0.1,0.5,0.9), data = Orthodont)

# Predict (y - Xb)
predict(fitOi.lqmm, level = 0)

# Predict (y - Xb - Zu)
predict(fitOi.lqmm, level = 1)

# 95% confidence intervals
predint(fitOi.lqmm, level = 0, alpha = 0.05)
```

---

print.lqm *Print LQM Objects*

Description

Print an object generated by lqm or lqm.counts.

Usage

```
## S3 method for class 'lqm'
print(x, digits = max(6, getOption("digits")), ...)
```

Arguments

x	an lqm or lqm.counts object.
digits	a non-null value for digits specifies the minimum number of significant digits to be printed in values.
...	not used.

Author(s)

Marco Geraci

See Also

lqm, lqm.counts

`print.lqmm` *Print an* `lqmm` *Object*

Description

Print an object generated by lqmm.

Usage

```
## S3 method for class 'lqmm'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	an lqmm object.
digits	a non-null value for digits specifies the minimum number of significant digits to be printed in values.
...	not used.

Author(s)

Marco Geraci

See Also

lqmm

print.summary.lqm *Print an* lqm *Summary Object*

Description

Print summary of an lqm object.

Usage

```
## S3 method for class 'summary.lqm'
print(x, ...)
```

Arguments

x	a summary.lqm object.
...	not used.

Author(s)

Marco Geraci

See Also

lqm, summary.lqm

print.summary.lqmm *Print an* lqmm *Summary Object*

Description

Print summary of an lqmm object.

Usage

```
## S3 method for class 'summary.lqmm'
print(x, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

x	a summary.lqmm object.
digits	a non-null value for digits specifies the minimum number of significant digits to be printed in values.
...	not used.

Author(s)

Marco Geraci

See Also

lqmm, summary.lqmm

---

ranef.lqmm *Extract Random Effects*

Description

This function computes random effects for a linear quantile mixed model.

Usage

```
## S3 method for class 'lqmm'
ranef(object, ...)
```

Arguments

object	an object of class lqmm.
...	not used.

Details

The prediction of the random effects is done via estimated best linear prediction (Geraci and Bottai, 2014). The generic function ranef is imported from the nlme package (Pinheiro et al, 2014).

Value

a data frame or a list of data frames of predicted random effects.

Author(s)

Marco Geraci

References

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479. doi: 10.1007/s11222-013-9381-9.

Pinheiro J, Bates D, DebRoy S, Sarkar D and R Core Team (2014). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-117, https://CRAN.R-project.org/package=nlme.

See Also

lqmm, coef.lqmm

residuals.lqm *Residuals from an LQM Objects*

Description

This function computes the residuals from a fitted linear quantile model.

Usage

```
## S3 method for class 'lqm'
residuals(object, ...)
```

Arguments

object	an lqm or lqm.counts object.
...	not used.

Value

a vector or matrix of residuals.

Author(s)

Marco Geraci

See Also

lqm, lqm.counts, predict.lqm, coef.lqm

residuals.lqmm *Residuals from an* lqmm *Object*

Description

The residuals at level 0 correspond to population residuals (based only on the fixed effects estimates). The residuals at level 1 are obtained by adding the best linear predictions of the random effects to the predictions at level 0 and the subtracting these from the model response.

Usage

```
## S3 method for class 'lqmm'
residuals(object, level = 0, ...)
```

Arguments

object	an lqmm object.
level	an optional integer vector giving the level of grouping to be used in obtaining the predictions. Level zero corresponds to the population residuals.
...	not used.

Value

a matrix of residuals.

Author(s)

Marco Geraci

References

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479. doi: 10.1007/s11222-013-9381-9.

See Also

lqmm, predict.lqmm, coef.lqmm, ranef.lqmm,

---

summary.boot.lqm *Summary for a* boot.lqm *Object*

Description

Summary method for class boot.lqm.

Usage

```
## S3 method for class 'boot.lqm'
summary(object, alpha = 0.05, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

object	an object of class lqm.
alpha	numeric value for the interval confidence level (1-alpha).
digits	a non-null value for digits specifies the minimum number of significant digits to be printed in values.
...	not used.

Author(s)

Marco Geraci

See Also

boot.lqm, lqm,

---

summary.boot.lqmm *Summary for a* boot.lqmm *Object*

Description

This function gives a summary of a botstrapped lqmm object

Usage

```
## S3 method for class 'boot.lqmm'
summary(object, alpha = 0.05, digits = max(3, getOption("digits") - 3), ...)
```

Arguments

object	an object of class lqmm.
alpha	numeric value for the interval confidence level (1-alpha).
digits	a non-null value for digits specifies the minimum number of significant digits to be printed in values.
...	not used.

Author(s)

Marco Geraci

References

Geraci M and Bottai M (2014). Linear quantile mixed models. Statistics and Computing, 24(3), 461–479. doi: 10.1007/s11222-013-9381-9.

See Also

boot.lqmm, lqmm,

summary.lqm *Summary for an* lqm *Object*

Description

Summary method for class lqm.

Usage

```
## S3 method for class 'lqm'
summary(object, method = "boot", alpha = 0.05, covariance = FALSE, ...)
```

Arguments

object	an object of class lqm
method	specifies the method used to compute standard errors: "boot" for bootstrap (default), "nid" for large sample approximations under *nid* assumptions.
alpha	significance level.
covariance	logical flag. If TRUE the covariance matrix of the bootstrap estimates is provided.
...	see boot.lqm for additional arguments.

Details

print.summary.lqm formats the coefficients, standard errors, etc. and additionally gives 'significance stars'.

Value

an object of class summary.lqm. The function summary.lqm computes and returns a list of summary statistics of the fitted linear quantile mixed model given in object, using the components (list elements) from its argument, plus

Cov	the covariance matrix obtained from the bootstrapped estimates (if covariance = TRUE).
tTable	a matrix with estimates, standard errors, etc.

Author(s)

Marco Geraci

Source

The code for the "nid" method has been adapted from the function summary.rq in package quantreg. It depends on the function bandwidth.rq.

Roger Koenker (2016). quantreg: Quantile Regression. R package version 5.29. https://CRAN.R-project.org/package=quantreg

See Also

print.summary.lqm lqm

Examples

```
set.seed(12356)
n <- 200
p <- 1:3/4
test <- data.frame(x = runif(n,0,1))
test$y <- 30 + test$x + rnorm(n)
fit.lqm <- lqm(y ~ x, data = test, tau = p)
summary(fit.lqm, R = 50)
```

---

summary.lqmm *Summary for an* lqmm *Object*

Description

Summary method for class lqmm.

Usage

```
## S3 method for class 'lqmm'
summary(object, method = "boot", alpha = 0.05, covariance = FALSE, ...)
```

Arguments

object	an object of class lqmm.
method	specifies the method used to compute standard errors. Currently, only the bootstrap method ("boot") is available.
alpha	significance level.
covariance	logical flag. If TRUE the covariance matrix of the bootstrap estimates is provided.
...	see boot.lqmm for additional arguments.

Details

print.summary.lqmm formats the coefficients, standard errors, etc. and additionally gives 'significance stars'.

Value

an object of class `summary.lqmm`. The function `summary.lqmm` computes and returns a list of summary statistics of the fitted linear quantile mixed model given in `object`, using the components (list elements) from its argument, plus

`Cov`	the covariance matrix obtained from the bootstrapped estimates (if `covariance = TRUE`).
`tTable`	a matrix with estimates, standard errors, etc.
`B`	the matrix of all bootstrapped parameters.

Author(s)

Marco Geraci

See Also

`print.summary.lqmm` `lqmm`

Examples

```
data(Orthodont)
fitOi.lqmm <- lqmm(distance ~ age, random = ~ 1, group = Subject,
tau = c(0.1,0.5,0.9), data = Orthodont)
summary(fitOi.lqmm)
```

VarCorr.lqmm *Extract Variance-Covariance Matrix*

Description

This function extracts the variance-covariance matrix of the random effects from a fitted `lqmm` object.

Usage

```
## S3 method for class 'lqmm'
VarCorr(x, sigma = NULL, ...)
```

Arguments

`x`	an object of `class` "lqmm".
`sigma`	not used.
`...`	not used.

Details

This function returns the variance or the variance-covariance matrix of the random effects. It calls `covHandling` to manage the output of `lqmm.fit.gs` or `lqmm.fit.df`. A post-fitting approximation to the nearest positive (semi)definite matrix (Higham, 2002) is applied if necessary. The generic function `VarCorr` is imported from the `nlme` package (Pinheiro et al, 2014).

Author(s)

Marco Geraci

References

Higham N (2002). Computing the Nearest Correlation Matrix - A Problem from Finance. IMA Journal of Numerical Analysis, 22, 329-343.

Pinheiro J, Bates D, DebRoy S, Sarkar D and R Core Team (2014). nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-117, https://CRAN.R-project.org/package=nlme.

See Also

`lqmm` `coef.lqmm`

Index

∗ **asymmetric Laplace distribution**
dal, 7
meanAL, 28
mleAL, 29
∗ **bootstrap**
boot, 3
extractBoot, 8
summary.boot.lqm, 38
summary.boot.lqmm, 39
summary.lqm, 40
summary.lqmm, 41
∗ **coefficients**
coef.lqm, 5
coef.lqmm, 6
ranef.lqmm, 36
VarCorr.lqmm, 42
∗ **control**
lqmControl, 18
lqmmControl, 26
∗ **covariance**
covHandling, 6
is.positive.definite, 10
make.positive.definite, 28
VarCorr.lqmm, 42
∗ **datasets**
labor, 10
Orthodont, 30
∗ **fitting**
lqm.fit.gs, 16
lqmControl, 18
lqmm.fit.df, 23
lqmm.fit.gs, 25
lqmmControl, 26
∗ **gaussian quadrature**
gauss.quad, 9
gauss.quad.prob, 9
∗ **maximum likelihood estimation**
meanAL, 28
mleAL, 29
∗ **models**
logLik.lqm, 11
logLik.lqmm, 12
∗ **positive definite**
is.positive.definite, 10
make.positive.definite, 28
∗ **prediction**
predict.lqm, 31
predict.lqmm, 32
∗ **print**
print.lqm, 33
print.lqmm, 34
print.summary.lqm, 35
print.summary.lqmm, 35
∗ **quantile regression**
lqm, 12
lqmm, 19
lqmm-package, 2
∗ **quantiles for counts**
lqm.counts, 14
∗ **random effects**
ranef.lqmm, 36
∗ **residuals**
residuals.lqm, 37
residuals.lqmm, 37
∗ **standard errors**
boot, 3
summary.lqm, 40
summary.lqmm, 41
∗ **summary**
print.summary.lqm, 35
print.summary.lqmm, 35
summary.boot.lqm, 38
summary.boot.lqmm, 39

AIC, *11, 12*
as.data.frame, *12*

boot, 3
boot.lqm, *31*, *39, 40*

boot.lqmm, *8*, *39*, *41*

class, *4*, *6*, *8*, *11–13*, *20*, *36*, *38–42*
coef.lqm, 5, *14*, *31*, *37*
coef.lqmm, 6, *22*, *33*, *36*, *38*, *43*
covHandling, *4*, 6, *43*

dal, 7, *29*

extractBoot, 8

F.lqm, *15*
formula, *12*, *15*, *19*

gauss.quad, 9
gauss.quad.prob, 9

invvarAL *(*meanAL*)*, 28
is.positive.definite, 10

labor, 10
lm, *4*
logLik.lqm, 11
logLik.lqmm, 12
lqm, *5*, *7*, *11*, 12, *17*, *19*, *22*, *27*, *31*, *33–35*, *37*, *39*, *41*
lqm.counts, *5*, 14, *31*, *33, 34*, *37*
lqm.fit.gs, *13*, 16
lqmControl, *13–16*, 18
lqmm, *6, 7*, *12*, 19, *24–27*, *33, 34*, *36*, *38, 39*, *42, 43*
lqmm-package, 2
lqmm.fit.df, *8*, *20, 21*, 23, *43*
lqmm.fit.gs, *8*, *20, 21*, 25, *43*
lqmmControl, *20, 21*, *24, 25*, 26

make.positive.definite, 28
meanAL, 28, *29*
mleAL, *29*, 29
model.frame, *21*
model.matrix.default, *13*, *15*

optim, *20*, *23*
optimize, *20*, *23*
options, *13*
Orthodont, 30

pal *(*dal*)*, 7
predict.lqm, *14*, 31, *37*
predict.lqmm, *22*, 32, *38*
predint *(*predict.lqmm*)*, 32
print.lqm, 33
print.lqmm, 34
print.summary.lqm, 35, *41*
print.summary.lqmm, 35, *42*

qal *(*dal*)*, 7

ral *(*dal*)*, 7
ranef *(*ranef.lqmm*)*, 36
ranef.lqmm, *33*, 36, *38*
residuals.lqm, *14*, *31*, 37
residuals.lqm.counts, *31*
residuals.lqmm, *22*, 37

summary.boot.lqm, 38
summary.boot.lqmm, 39
summary.lqm, *5*, *14*, *35*, 40
summary.lqmm, *6*, *22*, *36*, 41

varAL *(*meanAL*)*, 28
VarCorr *(*VarCorr.lqmm*)*, 42
VarCorr.lqmm, *7*, *21, 22*, *24, 25*, 42