Package ‘heavy’

February 20, 2015

Version 0.2-35
Date 2014-09-01
Title Package for robust estimation using heavy-tailed distributions
Author Felipe Osorio <felipe.osorios@usm.cl>
Maintainer Felipe Osorio <felipe.osorios@usm.cl>
Description This package contains functions to perform robust estimation considering
heavy-tailed distributions. Currently, the package includes linear regression, linear
mixed-effect models, multivariate location and scatter estimation, penalized splines
and random variate generation.
License GPL (>= 2)
URL http://heavy.mat.utfsm.cl/
LazyLoad yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-09-01 07:19:05

R topics documented:

creatinine ................................................................. 2
cyt ................................................................. 2
dental ............................................................... 3
erturns ............................................................. 4
heavy.control .......................................................... 4
heavy.family .......................................................... 5
heavyFit ............................................................ 6
heavyLm .............................................................. 7
heavyLme ........................................................... 9
heavyPS ............................................................. 10
life ................................................................. 12
rmCauchy ........................................................... 13
rmcnorm ............................................................. 14
rmnorm ............................................................. 15
**Description**

Data from a clinical trial on endogenous creatinine clearance of 34 male patients. Of the 34 male patients, two had no recorded the body weight, and four were missing the serum creatinine.

**Usage**

data(creatinine)

**Format**

A data frame with 34 observations on the following 4 variables.

- **WT**  body weight in kilograms.
- **SC**  serum creatinine in mg/deciliter.
- **Age**  age of the patients in years.
- **CR**  endogenous creatinine clearance.

**Source**


---

**Description**

Monthly returns of Concha y Toro vineyard collected over the period among the months of March 1990 to April 1999.

**Usage**

data(cyt)
An orthodontic measurement on several children over time
**etrends**

*Excess returns for Martin Marietta and American Can companies*

**Description**

Data from the Martin Marietta and American Can companies collected over a period of 5 years on a monthly basis.

**Usage**

```r
data(etrends)
```

**Format**

A data frame with 60 observations on the following 4 variables.

- **Date** the month in which the observations were collected.
- **am.can** excess returns from the American Can company.
- **m.marietta** excess returns from the Martin Marietta company.
- **CRSP** an index for the excess rate returns for the New York stock exchange.

**Source**


---

**heavy.control**

*Set control parameters*

**Description**

Allows users to set control parameters for routines available in heavy.

**Usage**

```r
heavy.control(maxIter = 4000, tolerance = 1e-6, fix.shape = FALSE, ndraws = 8000, algorithm = c("EM", "NEM"), ncycles = 5)
```
Arguments

- `maxIter` maximum number of iterations. The default is 4000.
- `tolerance` the relative tolerance in the iterative algorithm.
- `fix.shape` whether the shape parameters should be kept fixed in the fitting processes. The default is `fix.shape = FALSE`. Currently, only is available for the Student-t and slash distributions.
- `ndraws` independent draws for Monte Carlo integration. The default is 8000.
- `algorithm` character string specifying the algorithm to use. The default algorithm is an EM algorithm. Other possible value is "NEM" for the nested EM algorithm.
- `ncycles` number of cycles used in the nested EM algorithm, not used if `algorithm="EM"`. The default is 5.

Value

A list of control arguments to be used in a call to the fitter (i.e., `heavyFit`, `heavylm`, `heavylme` or `heavyps` functions).

A call to `heavy.control` can be used directly in the control argument of the call to the fitter.

See Also

- `heavyFit`, `heavylm`, `heavylme`, `heavyps`

Examples

```r
ctrl <- heavy.control(maxIter = 200, tol = 1e-04, ndraws = 9000)
data(dental)
heavylme(distance ~ age * Sex, random = ~ age, groups = ~ Subject,
          data = dental, control = ctrl)
```

Description

Provide a convenient way to specify the details of the models used by the fitter functions. (i.e., `heavyFit`, `heavylm`, `heavylme` and `heavyps` functions).

Usage

```r
normal()
Cauchy()
Student(df = 4)
slash(df = 2)
contaminated(epsilon = 0.05, vif = 0.25)
```
Arguments

- `df`: degrees of freedom for Student-t and slash distributions.
- `epsilon`, `vif`: percentage of outliers contamination and variance inflation factor.

Details

`heavy.family` is a generic function to create information about the selected heavy-tailed distribution which is passed to the estimation algorithm.

See Also

`heavyFit`, `heavyLm`, `heavyLme`, `heavyPS`

Examples

```r
CauchyFmly <- Student(df = 1)
CauchyFmly
```

**Description**

Estimates the center and scatter matrix assuming that the data came from a multivariate heavy-tailed distribution. This provides some degree of robustness to outliers without giving a high breakdown point.

**Usage**

```r
heavyFit(x, data, family = Student(df = 4), subset, na.action, control)
```

**Arguments**

- `x`: a formula or a numeric matrix or an object that can be coerced to a numeric matrix.
- `data`: an optional data frame (or similar: see `model.frame`), used only if `x` is a formula. By default the variables are taken from `environment(formula)`.
- `family`: a description of the error distribution to be used in the model. By default the Student-t distribution with 4 degrees of freedom is considered.
- `subset`: an optional expression indicating the subset of the rows of data that should be used in the fitting process.
- `na.action`: a function that indicates what should happen when the data contain NAs.
- `control`: a list of control values for the estimation algorithm to replace the default values returned by the function `heavy.control`. 
Value

A list with class "heavyFit" containing the following components:

- call: a list containing an image of the heavyFit call that produced the object.
- family: the heavyfamily object used, with the estimated shape parameters (if requested).
- center: final estimate of the location vector.
- Scatter: final estimate of the scale matrix.
- logLik: the log-likelihood at convergence.
- numIter: the number of iterations used in the iterative algorithm.
- weights: estimated weights corresponding to the assumed heavy-tailed distribution.
- distances: estimated squared Mahalanobis distances.
- acov: asymptotic covariance matrix of the center estimates.

References


See Also
cov

Examples

data(creatinine)
fit <- heavyFit(~ log(wt) + log(sc) + log(140 - Age) + log(CR), data = creatinine,
  family = Student(df = 4))
summary(fit)

Description

This function is used to fit linear models considering heavy-tailed errors.

Usage

heavylm(formula, data, family = Student(df = 4), subset, na.action,
  control, model = TRUE, x = FALSE, y = FALSE, contrasts = NULL)
Arguments

formula an object of class "formula": a symbolic description of the model to be fitted.
data an optional 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 heavyLm is called.
family a description of the error distribution to be used in the model. By default the Student-t distribution with 4 degrees of freedom is considered.
subset an optional expression indicating the subset of the rows of data that should be used in the fitting process.
n.a.action a function that indicates what should happen when the data contain NAs.
control a list of control values for the estimation algorithm to replace the default values returned by the function heavy.control.
model, x, y logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response) are returned.
contrasts an optional list. See the contrasts.arg of model.matrix.default.

Value

An object of class heavyLm representing the linear model fit. Generic functions print and summary, show the results of the fit.

The following components must be included in a legitimate heavyLm object.
call a list containing an image of the heavyLm call that produced the object.
family the heavy.family object used, with the estimated shape parameters (if requested).
coefficients final estimate of the coefficients vector.
sigma2 final scale estimate of the random error.
fitted.values the fitted mean values.
residuals the residuals, that is response minus fitted values.
logLik the log-likelihood at convergence.
numIter the number of iterations used in the iterative algorithm.
weights estimated weights corresponding to the assumed heavy-tailed distribution.
distances squared of scaled residuals.
acov asymptotic covariance matrix of the coefficients estimates.

References


Examples

data(ereturns)
fit <- heavyLm(m.marietta ~ CRSP, data = ereturns, family = Student(df = 5))
summary(fit)
Description

This function fits a linear mixed-effects model under heavy-tailed errors using the formulation described in Pinheiro et al. (2001).

Usage

```R
heavylme(fixed, random, groups, data, family = Student(df = 4),
         subset, na.action, control)
```

Arguments

- `fixed`: a two-sided linear formula object describing the fixed-effects part of the model, with the response on the left of a `~` operator and the terms, separated by `+` operators.
- `random`: a one-sided formula of the form `~x1+...+xn` specifying the model for the random effects.
- `groups`: a one-sided formula for specifying the grouping variable.
- `data`: an optional data frame containing the variables named in `fixed`, `random` and `groups`. By default the variables are taken from the environment from which `heavylme` is called.
- `family`: a description of the error distribution to be used in the model. By default the Student-t distribution with 4 degrees of freedom is considered.
- `subset`: an optional expression indicating the subset of the rows of data that should be used in the fitting process.
- `na.action`: a function that indicates what should happen when the data contain NAs.
- `control`: a list of control values for the estimation algorithm to replace the default values returned by the function `heavylme.control`.

Value

An object of class `heavylme` representing the linear mixed-effects model fit. Generic function `print` and `summary`, show the results of the fit.

The following components must be included in a legitimate `heavylme` object.

- `lmeData`: an object representing a list of mixed-effects model components.
- `call`: a list containing an image of the `heavylme` call that produced the object.
- `family`: the `heavylme.family` object used, with the estimated shape parameters (if requested).
- `coefficients`: final estimate of the fixed effects.
- `theta`: final estimate of the scale parameters associated to the random effects.
scale final scale estimate of the random error.
logLik the log-likelihood at convergence.
umIter the number of iterations used in the iterative algorithm.
ranef a matrix with the estimated random effects.
weights estimated weights corresponding to the assumed heavy-tailed distribution.
distances estimated squared Mahalanobis distances.
Fitted a data frame with the "marginal" and "conditional" fitted values as columns.
Resid a data frame with the "marginal" and "conditional" residuals as columns.

References


Examples

data(dental)
fit <- heavylme(distance ~ age * Sex, random = ~ age, groups = ~ Subject,
data = dental, family = Student(df = 4))
summary(fit)

---

heavyPS

Fit a penalized spline under heavy-tailed distributions

Description

Fits a penalized spline to the supplied data.

Usage

heavyPS(x, y, family = Student(df = 4), nseg = 20, deg = 3, ord = 2,
lambda = 1, method = "gcv", ngrid = 200, control)

Arguments

x, y vectors giving the coordinates of the points in the scatter plot. Missing values are deleted.
family a description of the error distribution to be used in the model. By default the Student-t distribution with 4 degrees of freedom is considered.
nseg number of segments used to divide the domain, this information is required to construct the sequence of knots. Default value is 20.
deg the degree of the spline transformation. Must be a nonnegative integer. The default value is 3. The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful.
ord the order of the roughness penalty. Default value is 2.

lambda specifies the smoothing parameter for the fit. It is fixed if method="none". If method="gcv" then weighted generalized cross validation is used to choose an "optimal" smoothing parameter. The default value is 1.

method the method for choosing the smoothing parameter lambda. If method="none", then lambda is 'fixed'. If method="gcv" then the smoothing parameter is chosen automatically using the weighted generalized cross validation criterion.

ngrid number of elements in the grid used to compute the smoother. Only required to plot the fitted P-spline.

control a list of control values for the estimation algorithm to replace the default values returned by the function heavy.control.

Value

an object of class heavyPS representing the fitted model. Generic functions print and summary, show the results of the fit.

The following components must be included in a legitimate heavyPS object.

call a list containing an image of the heavyPS call that produced the object.
design a list containing the B-spline basis matrix, the triangular factor of the penalty matrix and a numeric vector of knot positions with non-decreasing values.

method one of "gcv" or "none", depending on the fitting criterion used.
family the heavy.family object used in the fitting process.
coefficients final estimate of the coefficients vector.
scale final scale estimate of the random error.
lambda estimated smoothing parameter for the model (if requested).
fitted.values fitted model predictions of expected value for each datum.
residuals the residuals for the fitted model.
ploglik the penalized log-likelihood at convergence.
edf the effective number of parameters.
gcv the minimized smoothing parameter selection score (weighted GCV).
pen the penalty term at convergence.
umIter the number of iterations used in the iterative algorithm.
weights estimated weights corresponding to the assumed heavy-tailed distribution.
distances squared of scaled residuals.
xgrid grid of x-values used to fit the P-spline.
ygrid estimated curve on the x-grid, required to plot the fitted P-spline.
shape estimated shape parameters, only available if requested.
References

Examples

```r
data(life)
x <- life$income
y <- life$life
fit <- heavyPS(x, y, family = Student(df = 5))
summary(fit)
par(pty = "s")
plot(x, y, xlab = "Per Capita Income", ylab = "Life Expectancy")
lines(fit$xgrid, fit$ygrid)
```

---

**life**

*Data on per-capita income and life expectancy*

Description
The data represent life expectancies and per capita income in 1974 dollars for 101 countries in the early 1970's. The data were originally published in the New York Times (September, 28, 1975, p. E-3), they also appear on Leinhardt and Wasserman (1979) and Regression Analysis by Ashish Sen and Muni Srivastava.

Usage

```r
data(life)
```

Format
A data frame with 101 observations on the following 3 variables.

- **Country** character string, containing names of the countries.
- **life** life expectancy, in years for 1974.
- **income** income per capita in 1974 US dollars.

Source
Description

Random number generation from the multivariate Cauchy distribution.

Usage

```r
rmCauchy(n = 1, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)))
```

Arguments

- `n` the number of samples requested
- `center` a vector giving the positions of each variable
- `Scatter` a positive-definite dispersion matrix

Details

The function `rmCauchy` is an interface to C routines, which make calls to subroutines from LAPACK. The matrix decomposition is internally done using the Cholesky decomposition. If `Scatter` is not non-negative definite then there will be a warning message.

Value

- If `n = 1` a vector of the same length as `center`, otherwise a matrix of `n` rows of random vectors.

References


See Also

- `rcauchy`

Examples

```r
# dispersion parameters
Scatter <- matrix(c(10,3,3,2), ncol = 2)
Scatter

# generate the sample
y <- rmCauchy(n = 1000, Scatter = Scatter)

# scatterplot of a random bivariate Cauchy sample with center
# vector zero and scale matrix 'Scatter'
par(pty = "s")
plot(y, xlab = "", ylab = "")
title("bivariate Cauchy sample", font.main = 1)
```
Description
Random number generation from the multivariate contaminated normal distribution.

Usage
rmcnorm(n = 1, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)),
epsilon = 0.05, vif = 0.25)

Arguments
- n: the number of samples requested
- center: a vector giving the locations of each variable
- Scatter: a positive-definite dispersion matrix
- epsilon, vif: percentage of outliers contamination and variance inflation factor

Details
The function `rmcnorm` is an interface to C routines, which make calls to subroutines from LAPACK. The matrix decomposition is internally done using the Cholesky decomposition. If `Scatter` is not non-negative definite then there will be a warning message.

Value
If \( n = 1 \) a vector of the same length as `center`, otherwise a matrix of \( n \) rows of random vectors.

References

Examples
```r
# dispersion parameters
Scatter <- matrix(c(10,3,3,2), ncol = 2)
Scatter

# generate the sample
y <- rmcnorm(n = 1000, Scatter = Scatter, epsilon = .05, vif = .25)

# scatterplot of a random bivariate contaminated normal sample
# with center vector zero and scale matrix 'Scatter'
par(pty = "s")
plot(y, xlab = "", ylab = "")
title("bivariate contaminated normal sample", font.main = 1)
```
Description
Random number generation from the multivariate normal (Gaussian) distribution.

Usage
\texttt{rmnorm(n = 1, mean = rep(0, nrow(Sigma)), Sigma = diag(length(mean)))}

Arguments
- \texttt{n} the number of samples requested
- \texttt{mean} a vector giving the means of each variable
- \texttt{Sigma} a positive-definite covariance matrix

Details
The function \texttt{rmnorm} is an interface to C routines, which make calls to subroutines from LAPACK. The matrix decomposition is internally done using the Cholesky decomposition. If \texttt{Sigma} is not non-negative definite then there will be a warning message.

Value
If \( n = 1 \) a vector of the same length as \texttt{mean}, otherwise a matrix of \( n \) rows of random vectors.

References

See Also
\texttt{rnorm}

Examples
\begin{verbatim}
# covariance parameters
Sigma <- matrix(c(10,3,3,2), ncol = 2)
Sigma

# generate the sample
y <- rmnorm(n = 1000, Sigma = Sigma)
var(y)

# scatterplot of a random bivariate normal sample with mean # vector zero and covariance matrix 'Sigma'
par(pty = "s")
plot(y, xlab = "", ylab = "")
title("bivariate normal sample", font.main = 1)
\end{verbatim}
**Description**

Random number generation from the multivariate Slash distribution.

**Usage**

```r
rmslash(n = 1L, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)), df = 2)
```

**Arguments**

- `n`  
  the number of samples requested
- `center`  
  a vector giving the locations of each variable
- `Scatter`  
  a positive-definite dispersion matrix
- `df`  
  degrees of freedom (positive, maybe non-integer)

**Details**

The function `rmslash` is an interface to C routines, which make calls to subroutines from LAPACK. The matrix decomposition is internally done using the Cholesky decomposition. If `Scatter` is not non-negative definite then there will be a warning message.

**Value**

If `n = 1` a vector of the same length as `center`, otherwise a matrix of `n` rows of random vectors.

**References**


**Examples**

```r
# dispersion parameters
Scatter <- matrix(c(10, 3, 3, 2), ncol = 2)
Scatter

# generate the sample
y <- rmslash(n = 1000, Scatter = Scatter)

# scatterplot of a random bivariate slash sample with center
# vector zero and scale matrix 'Scatter'
par(pty = "s")
plot(y, xlab = "", ylab = "")
title("bivariate slash sample (df = 2)", font.main = 1)
```
Multivariate Student-t Random Deviates

Description

Random number generation from the multivariate Student-t distribution.

Usage

```r
rmt(n = 1L, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)), df = TI)
```

Arguments

- `n`: the number of samples requested
- `center`: a vector giving the locations of each variable
- `Scatter`: a positive-definite dispersion matrix
- `df`: degrees of freedom (positive, maybe non-integer)

Details

The function `rmt` is an interface to C routines, which make calls to subroutines from LAPACK. The matrix decomposition is internally done using the Cholesky decomposition. If `Scatter` is not non-negative definite then there will be a warning message.

Value

If `n = 1` a vector of the same length as `center`, otherwise a matrix of `n` rows of random vectors.

References


See Also

- `rt`

Examples

```r
# dispersion parameters
Scatter <- matrix(c(10,3,3,2), ncol = 2)
Scatter

# generate the sample
y <- rmt(n = 1000, Scatter = Scatter)

# scatterplot of a random bivariate Student-t sample with center
# vector zero and scale matrix 'Scatter'
```
rsphere

Generation of Deviates Uniformly Located on a Spherical Surface

Description
Random vector generation uniformly on the sphere.

Usage
rsphere(n = 1, p = 2)

Arguments
- **n**: the number of samples requested
- **p**: dimension of the unitary sphere

Details
The function `rsphere` is an interface to C routines, which make calls to subroutines from BLAS.

Value
If \( n = 1 \) a \( p \)-dimensional vector, otherwise a matrix of \( n \) rows of random vectors.

References

See Also
- `runif`

Examples
```r
# generate the sample
z <- rsphere(n = 200)

# scatterplot of a random sample of 200 points uniformly distributed
# on the unit circle
par(pty = "s")
plot(z, xlab = "x", ylab = "y")
title("200 points on the circle", font.main = 1)
```
Index

*Topic **datasets**
  creatinine, 2
cyt, 2
dental, 3
ereturns, 4
life, 12

*Topic **distribution**
rmCauchy, 13
rmcnorm, 14
rmnorm, 15
rmslash, 16
rmt, 17
rsphere, 18

*Topic **models**
heavy.control, 4
heavy.family, 5
heavyLm, 7
heavyLme, 9

*Topic **multivariate**
heavyFit, 6
rmCauchy, 13
rmcnorm, 14
rmnorm, 15
rmslash, 16
rmt, 17
rsphere, 18

*Topic **regression**
heavyLm, 7

*Topic **smooth**
heavyPS, 10

Cauchy (heavy.family), 5
contaminated (heavy.family), 5
cov, 7
creatinine, 2
cyt, 2
dental, 3
ereturns, 4

heavy.control, 4, 6, 8, 9, 11
heavy.family, 5, 7–9, 11
heavyFit, 5, 6, 6
heavyLm, 5, 6, 7
heavyLme, 5, 6, 9
heavyPS, 5, 6, 10

life, 12
model.frame, 6
model.matrix.default, 8
normal (heavy.family), 5
rcauchy, 13
rmCauchy, 13
rmcnorm, 14
rmnorm, 15
rmslash, 16
rmt, 17
rnorm, 15
rsphere, 18
rt, 17
runif, 18
slash (heavy.family), 5
Student (heavy.family), 5