

Package ‘heavy’

February 13, 2018

Title Robust Estimation Using Heavy-Tailed Distributions

Version 0.38.19

Date 2018-02-13

Author Felipe Osorio

Maintainer Felipe Osorio <felipe.osorios@usm.cl>

Description Functions to perform robust estimation considering heavy-tailed distributions. Currently, the package includes routines for linear regression, linear mixed-effect models, multivariate location and scatter estimation, multivariate regression, penalized splines, random variate generation and some support functions.

URL <http://heavy.mat.utfsm.cl>

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-02-13 22:04:39 UTC

R topics documented:

creatinine	2
cyt	3
dental	3
dialyzer	4
ereturns	4
heavy.control	5
heavy.family	6
heavyFit	6
heavyLm	8
heavyLm.fit	10
heavyLme	11
heavyPS	12
housePrices	14
life	15

pgamma.deriv	15
rmCauchy	16
rmcnorm	17
rmnorm	18
rmslash	19
rmt	21
rsphere	22
tgamma	23

Index	25
--------------	-----------

creatinine	<i>Clinical trial on endogenous creatinine clearance</i>
------------	--

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

Liu, C., and Rubin, D.B. (1995). ML estimation of the t distribution using EM and its extensions, ECM and ECME. *Statistica Sinica* **5**, 19-39.

cyt	<i>Concha y Toro vineyards</i>
-----	--------------------------------

Description

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

Usage

```
data(cyt)
```

Format

A data frame with 110 observations on the following 3 variables.

Date the month in which the observations were collected.

IPSA monthly returns for the Chilean market index.

CyT monthly returns, adjusted by equity variations, of Concha y Toro Vineyards.

Source

Osorio, F., and Galea, M. (2006). Detection of a change-point in student-t linear regression models. *Statistical Papers* **47**, 31-48.

dental	<i>An orthodontic measurement on several children over time</i>
--------	---

Description

The dental data set introduced by Potthoff and Roy (1964), where dental measurements were made on 11 girls and 16 boys at ages 8, 10, 12 and 14. The response variable was the distance, in millimeters, from the center of pituitary to the pterygomaxillary fissure.

Usage

```
data(dental)
```

Format

data frame with 108 rows and 4 columns.

Source

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

dialyzer

Ultrafiltration measurements for three centres

Description

Data from a study to evaluate the *in vivo* ultrafiltration characteristics of a group of hollow fiber dialyzers. The dialyzers were evaluated in three centres with each centre using a different type of dialysate delivery system. The data set was introduced by Vonesh and Carter (1987) consists of four ultrafiltration rate measurements (y_1 , y_2 , y_3 and y_4).

Usage

```
data(dialyzer)
```

Format

A data frame with 40 rows and 5 columns.

Source

Munoz-Pichardo, J.M., Moreno-Rebollo, J.L., Enguix-Gonzalez, A., and Pascual-Acosta, A. (2008). Influence measures on profile analysis with elliptical data through Frechet's metric. *Metrika* **68**, 111-127.

Vonesh, E.F., and Carter, R.L. (1987). Efficient inference for random-coefficient growth curve models with unbalanced data. *Biometrics* **43**, 617-628.

ereturns

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

```
data(ereturns)
```

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

Butler, R.J., McDonald, J.B., Nelson, R.D., and White, S.B. (1990). Robust and partially adaptive estimation of regression models. *The Review of Economics and Statistics* **72**, 321-327.

heavy.control	<i>Set control parameters</i>
---------------	-------------------------------

Description

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

Usage

```
heavy.control(maxIter = 2000, tolerance = 1e-6, fix.shape = FALSE,
             ndraws = 500, algorithm = c("EM", "NEM"), ncycles = 5)
```

Arguments

maxIter	maximum number of iterations. The default is 2000.
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 <code>fix.shape = FALSE</code> . Currently, only is available for the Student-t and slash distributions.
ndraws	independent draws for Monte Carlo integration. The default is 500.
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 <code>algorithm="EM"</code> . 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

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

heavy.family	<i>Family objects for heavy-tailed distributions</i>
--------------	--

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

```
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

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

heavyFit	<i>Estimation of Location and Scatter using multivariate heavy-tailed distributions</i>
----------	---

Description

Estimates the center vector 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

```
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 <code>environment(formula)</code> .
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 heavy.family 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

- Kent, J.T., Tyler, D.E., and Vardi, Y. (1994). A curious likelihood identity for the multivariate t-distribution. *Communications in Statistics - Simulation and Computation* **23**, 441-453.
- Lange, K., Little, R.J.A., and Taylor, J.M.G. (1989). Robust statistical modeling using the t distribution. *Journal of the American Statistical Association* **84**, 881-896.
- Little, R.J.A. (1988). Robust estimation of the mean and covariance matrix from data with missing values. *Applied Statistics* **37**, 23-38.

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)
```

heavyLm

*(Multivariate) linear regression under heavy-tailed distributions***Description**

This function is used to fit linear models considering heavy-tailed errors. It can be used to carry out univariate or multivariate regression.

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.
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 .
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 .

Details

Models for heavyLm are specified symbolically (for additional information see the "Details" section from [lm](#) function). If response is a matrix, then a multivariate linear model is fitted.

Value

An object of class "heavyLm" or "heavyMLm" for multiple responses which represents the fitted model. Generic functions `print` and `summary`, show the results of the fit.

The following components must be included in a legitimate "heavyLm" object.

<code>call</code>	a list containing an image of the heavyLm call that produced the object.
<code>family</code>	the <code>heavy.family</code> object used, with the estimated shape parameters (if requested).
<code>coefficients</code>	final estimate of the coefficients vector.
<code>sigma2</code>	final scale estimate of the random error (only available for univariate regression models).
<code>Sigma</code>	estimate of scatter matrix for each row of the response matrix (only available for objects of class "heavyMLm").
<code>fitted.values</code>	the fitted mean values.
<code>residuals</code>	the residuals, that is response minus fitted values.
<code>logLik</code>	the log-likelihood at convergence.
<code>numIter</code>	the number of iterations used in the iterative algorithm.
<code>weights</code>	estimated weights corresponding to the assumed heavy-tailed distribution.
<code>distances</code>	squared of scaled residuals or Mahalanobis distances.
<code>acov</code>	asymptotic covariance matrix of the coefficients estimates.

References

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1980). Iteratively reweighted least squares for linear regression when errors are Normal/Independent distributed. In P.R. Krishnaiah (Ed.), *Multivariate Analysis V*, p. 35-57. North-Holland.

Lange, K., and Sinsheimer, J.S. (1993). Normal/Independent distributions and their applications in robust regression. *Journal of Computational and Graphical Statistics* **2**, 175-198.

Examples

```
# univariate linear regression
data(ereturns)
fit <- heavyLm(m.marietta ~ CRSP, data = ereturns, family = Student(df = 5))
summary(fit)

# multivariate linear regression
data(dialyzer)
fit <- heavyLm(cbind(y1,y2,y3,y4) ~ -1 + centre, data = dialyzer, family = slash(df = 4))
fit
```

heavyLm.fit

*Fitter functions for linear models under heavy-tailed distributions***Description**

These are basic computing functions called by [heavyLm](#) used to fit linear models considering heavy-tailed errors. These should usually *not* be used directly unless by experienced users.

Usage

```
heavyLm.fit(x, y, family = Student(df = 4), control = heavy.control())
```

```
heavyMLm.fit(x, y, family = Student(df = 4), control = heavy.control())
```

Arguments

x	design matrix of dimension $n * p$.
y	vector of observations of length n , or a matrix with n rows.
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.
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 components	
family	the heavy.family object used, with the estimated shape parameters (if requested).
coefficients	p vector or a matrix with p rows.
sigma2	scale estimate of the random error (only available for univariate regression models).
Sigma	estimate of scatter matrix for each row of the response matrix (only available for objects of class "heavyMLm").
residuals	n vector or matrix
fitted.values	n vector or matrix
weights	estimated weights corresponding to the assumed heavy-tailed distribution.
distances	squared of scaled residuals or Mahalanobis distances.
acov	asymptotic covariance matrix of the coefficients estimates.
logLik	the log-likelihood at convergence.

See Also

[heavyLm](#) which you should use for multivariate or univariate linear regression under heavy-tailed distributions unless you know better.

Examples

```
# univariate linear regression
data(ereturns)
x <- cbind(1, ereturns$CRSP)
colnames(x) <- c("Intercept", "CRSP")
y <- ereturns$m.marietta
z <- heavyLm.fit(x = x, y = y)

# multivariate linear regression
data(dialyzer)
y <- as.matrix(dialyzer[,1:4])
n <- nrow(y)
x <- matrix(1, nrow = n, ncol = 1) # a vector of ones
z <- heavyMLm.fit(x = x, y = y)
```

heavyLme

*Linear mixed-effects models under heavy-tailed distributions***Description**

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

Usage

```
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 \sim operator and the terms, separated by + operators.
random	a one-sided formula of the form $\sim x_1 + \dots + x_n$ 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 group. By default the variables are taken from the environment from which heavy.lme 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 heavy.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.

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

References

Pinheiro, J.C., Liu, C., and Wu, Y.N. (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

```
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 = c("GCV", "none"), ngrid = 200, control)
```

Arguments

<code>x, y</code>	vectors giving the coordinates of the points in the scatter plot. Missing values are deleted.
<code>family</code>	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.
<code>nseg</code>	number of segments used to divide the domain, this information is required to construct the sequence of knots. Default value is 20.
<code>deg</code>	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.
<code>ord</code>	the order of the roughness penalty. Default value is 2.
<code>lambda</code>	specifies the smoothing parameter for the fit. It is fixed if <code>method="none"</code> . If <code>method="GCV"</code> then weighted generalized cross validation is used to choose an "optimal" smoothing parameter. The default value is 1.
<code>method</code>	the method for choosing the smoothing parameter <code>lambda</code> . If <code>method="none"</code> , then <code>lambda</code> is 'fixed'. If <code>method="GCV"</code> (the default) then the smoothing parameter is chosen automatically using the weighted generalized cross validation criterion.
<code>ngrid</code>	number of elements in the grid used to compute the smoother. Only required to plot the fitted P-spline.
<code>control</code>	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.

<code>call</code>	a list containing an image of the <code>heavyPS</code> call that produced the object.
<code>design</code>	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.
<code>method</code>	one of "GCV" or "none", depending on the fitting criterion used.
<code>family</code>	the heavy.family object used in the fitting process.
<code>coefficients</code>	final estimate of the coefficients vector.
<code>scale</code>	final scale estimate of the random error.
<code>lambda</code>	estimated smoothing parameter for the model (if requested).
<code>fitted.values</code>	fitted model predictions of expected value for each datum.
<code>residuals</code>	the residuals for the fitted model.
<code>plogLik</code>	the penalized log-likelihood at convergence.
<code>edf</code>	the effective number of parameters.
<code>gcv</code>	the minimized smoothing parameter selection score (weighted GCV).

pen	the penalty term 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.
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

Eilers, P.H.C., and Marx, B.D. (1996). Flexible smoothing using B-splines and penalties (with discussion). *Statistical Science* **11**, 89-121.

Osorio, F. (2016). Influence diagnostics for robust P-splines using scale mixture of normal distributions. *Annals of the Institute of Statistical Mathematics* **68**, 589-619.

Examples

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

housePrices

Hedonic housing-price equation

Description

Harrison and Rubinfeld (1978) reported a study of house prices in the Boston Metropolitan area. Their principal goal was to assess the association of the air quality in the neighbourhood with the house price after taking account of other influences.

Usage

```
data(housePrices)
```

Format

data frame with 506 rows and 15 columns.

Source

Harrison, D., and Rubinfeld, D.L. (1978). Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management* **5**, 81-102.

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

```
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

Sen, A., and Srivastava, M. (1990). *Regression Analysis: Theory, Methods and Applications*. Springer-Verlag, New York.

Leinhardt, S., and Vasserman, S.S. (1979). Teaching regression: An exploratory approach. *The American Statistician* **33**, 196-203.

pgamma.deriv *Derivatives of the regularized incomplete gamma function*

Description

Computes the incomplete gamma function and its first and second derivatives with respect to 'shape' parameter.

Usage

```
pgamma.deriv(x, shape, scale, deriv = 0:2)
```

Arguments

x a numeric argument, $x > 0$.

shape, scale shape and scale parameters, must be positive.

deriv integer vector, order of the required derivatives.

Details

The regularized (standard) incomplete gamma function is given by

$$P(a, x) = \frac{1}{\Gamma(a)} \int_0^x t^{a-1} e^{-t} dt,$$

which is related with the CDF of Gamma distribution (see [pgamma](#)).

`pgamma.deriv` is a C translation of the original Fortran AS 187 subroutine by R.J. Moore.

Value

Function `pgamma.deriv` returns the value, first and second derivatives of the regularized incomplete gamma function with respect to 'shape' parameter.

References

Abramowitz, M., and Stegun, I.A. (1970). *Handbook of Mathematical Functions*. Dover, New York.

Moore, R.J. (1982). Algorithm AS 187: Derivatives of the incomplete gamma integral. *Applied Statistics* **31**, 330-335.

See Also

[pgamma](#), [psigamma](#)

Examples

```
x <- 3
pgamma.deriv(x, 1, 1, deriv = 0:2)

# only to check 1st element
pgamma(x, 1, 1)
```

 rmCauchy

Multivariate Cauchy Random Deviates

Description

Random number generation from the multivariate Cauchy distribution.

Usage

```
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

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

See Also

[rcauchy](#)

Examples

```
# 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)
```

 rmcnorm

Multivariate Contaminated Normal Random Deviates

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

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

Examples

```

# 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)

```

 rmnorm

Multivariate Normal Random Deviates

Description

Random number generation from the multivariate normal (Gaussian) distribution.

Usage

```
rmnorm(n = 1, mean = rep(0, nrow(Sigma)), Sigma = diag(length(mean)))
```

Arguments

n	the number of samples requested
mean	a vector giving the means of each variable
Sigma	a positive-definite covariance matrix

Details

The function `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 `Sigma` is not non-negative definite then there will be a warning message.

Value

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

References

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

See Also

[rnorm](#)

Examples

```
# 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)
```

rmslash

Multivariate Slash Random Deviates

Description

Random number generation from the multivariate Slash distribution.

Usage

```
rmslash(n = 1, 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

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

Examples

```
# 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)
```

rmt

Multivariate Student-t Random Deviates

Description

Random number generation from the multivariate Student-t distribution.

Usage

```
rmt(n = 1, center = rep(0, nrow(Scatter)), Scatter = diag(length(center)),
    df = 4)
```

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

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

See Also

[rt](#)

Examples

```
# 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'
```

```
par(pty = "s")
plot(y, xlab = "", ylab = "")
title("bivariate Student-t sample (df = 4)", font.main = 1)
```

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

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

See Also

[runif](#)

Examples

```
# 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)
```

tgamma	<i>The right truncated gamma distribution</i>
--------	---

Description

Density, distribution function, quantile function and random generation for the right truncated gamma distribution with shape (`shape`), scale (`scale`) parameters and right truncation point (`truncation`).

Usage

```
dtgamma(x, shape, scale = 1, truncation = 1, log = FALSE)
ptgamma(q, shape, scale = 1, truncation = 1, lower.tail = TRUE)
qtgamma(p, shape, scale = 1, truncation = 1, lower.tail = TRUE)
rtgamma(n, shape, scale = 1, truncation = 1)
```

Arguments

<code>x</code> , <code>q</code>	vector of quantiles.
<code>shape</code> , <code>scale</code>	shape and scale parameters, must be positive.
<code>truncation</code>	right truncation point, must be positive.
<code>log</code>	logical; if TRUE, the log-density is returned.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
<code>p</code>	vector of probabilities.
<code>n</code>	number of observations. If <code>length(n) > 1</code> , the length is taken to be the number of required deviates.

Details

If `scale` or `truncation` are not specified, they assume the default values.

The right truncated gamma distribution with shape a , scale b and right truncation point $t > 0$ has density

$$f(x) = \frac{b^a}{\gamma(a, bt)} \exp(-bx)x^{a-1}$$

con $x < t$ and $\gamma(a, b)$ denotes the incomplete gamma function (see Abramowitz and Stegun, 1970, pp. 260).

Value

`dtgamma`, `ptgamma`, and `qtgamma` are respectively the density, distribution function and quantile function of the right truncated gamma distribution. `rtgamma` generates random deviates from the right truncated gamma distribution.

The length of the result is determined by `n` for `rtgamma`, and is the maximum of the lengths of the numerical parameters for the other functions.

References

Abramowitz, M., and Stegun, I.A. (1970). *Handbook of Mathematical Functions*. Dover, New York.

Phillippe, A. (1997). Simulation of right and left truncated gamma distribution by mixtures. *Statistics and Computing* 7, 173-181.

See Also

[Distributions](#) for other standard distributions.

Examples

```
x <- seq(0, 2, by = 0.1)
y <- dtgamma(x, shape = 1, truncation = 1)
z <- dgamma(x, shape = 1) # standard gamma pdf
plot(x, z, type = "l", xlab = "x", ylab = "density", ylim = range(y, z), lty = 2)
lines(x, y)

x <- rtgamma(1000, shape = 1)
## Q-Q plot for the right truncated gamma data against true theoretical distribution:
qqplot(qtgamma(ppoints(1000), shape = 1), x, main = "Truncated Gamma Q-Q plot",
       xlab = "Theoretical quantiles", ylab = "Sample quantiles", font.main = 1)
abline(c(0,1), col = "red", lwd = 2)
```


Index

- *Topic **array**
 - heavyLm.fit, 10
- *Topic **datasets**
 - creatinine, 2
 - cyt, 3
 - dental, 3
 - dialyzer, 4
 - ereturns, 4
 - housePrices, 14
 - life, 15
- *Topic **distribution**
 - rmCauchy, 16
 - rmcnorm, 17
 - rmnorm, 18
 - rmslash, 19
 - rmt, 21
 - rsphere, 22
 - tgamma, 23
- *Topic **math**
 - pgamma.deriv, 15
- *Topic **models**
 - heavy.control, 5
 - heavy.family, 6
 - heavyLm, 8
 - heavyLme, 11
- *Topic **multivariate**
 - heavyFit, 6
 - rmCauchy, 16
 - rmcnorm, 17
 - rmnorm, 18
 - rmslash, 19
 - rmt, 21
 - rsphere, 22
- *Topic **regression**
 - heavyLm, 8
 - heavyLm.fit, 10
- *Topic **smooth**
 - heavyPS, 12
- Cauchy (heavy.family), 6
- contaminated (heavy.family), 6
- cov, 7
- creatinine, 2
- cyt, 3
- dental, 3
- dialyzer, 4
- Distributions, 24
- dtgamma (tgamma), 23
- ereturns, 4
- heavy.control, 5, 7, 8, 10, 11, 13
- heavy.family, 6, 7, 9, 10, 12, 13
- heavyFit, 5, 6, 6
- heavyLm, 5, 6, 8, 10
- heavyLm.fit, 10
- heavyLme, 5, 6, 11
- heavyMLm.fit (heavyLm.fit), 10
- heavyPS, 5, 6, 12
- housePrices, 14
- life, 15
- lm, 8
- model.frame, 7
- model.matrix.default, 8
- normal (heavy.family), 6
- pgamma, 16
- pgamma.deriv, 15
- psigamma, 16
- ptgamma (tgamma), 23
- qtgamma (tgamma), 23
- rcauchy, 17
- rmCauchy, 16
- rmcnorm, 17
- rmnorm, 18

rmslash, 19
rmt, 21
rnorm, 19
rsphere, 22
rt, 21
rtgamma (tgamma), 23
runif, 22

slash (heavy.family), 6
Student (heavy.family), 6

tgamma, 23