Package ‘relliptical’

May 9, 2026

Type Package

Title The Truncated Elliptical Family of Distributions

Version 1.4.0

Description

It provides a function for random number generation from members of the truncated multivariate elliptical family of distributions, including truncated versions of the Normal, Student-t, Pearson type VII, Slash, Logistic, and related distributions. Additional distributions can be specified by supplying the density generating function. The package also computes first- and second-order moments, including the covariance matrix, for selected distributions.
References used for this package: Galarza, C. E., Matos, L. A., Castro, L. M., & Lachos, V. H. (2022). Moments of the doubly truncated selection elliptical distributions with emphasis on the unified multivariate skew-t distribution. Journal of Multivariate Analysis, 189, 104944 <doi:10.1016/j.jmva.2021.104944>; Ho, H. J., Lin, T. I., Chen, H. Y., & Wang, W. L. (2012). Some results on the truncated multivariate t distribution. Journal of Statistical Planning and Inference, 142(1), 25-40 <doi:10.1016/j.jspi.2011.06.006>; Valeriano, K. A., Galarza, C. E., & Matos, L. A. (2023). Moments and random number generation for the truncated elliptical family of distributions. Statistics and Computing, 33(1), 32 <doi:10.1007/s11222-022-10200-4>.

License GPL (>= 2)

Encoding UTF-8

Imports FuzzyNumbers.Ext.2, matrixcalc, methods, Rcpp, Rdpack, Ryacas, stats

RdMacros Rdpack

RoxygenNote 7.3.3

LinkingTo RcppArmadillo, Rcpp

Suggests ggExtra, ggplot2, gridExtra

NeedsCompilation yes

Author Katherine A. L. Valeriano [aut, cre] (ORCID: <https://orcid.org/0000-0001-6388-4753>),
Larissa Avila Matos [ctb] (ORCID: <https://orcid.org/0000-0002-2635-0901>),
Christian Galarza Morales [ctb] (ORCID: <https://orcid.org/0000-0002-4818-6006>)

Maintainer Katherine A. L. Valeriano <katandreina@gmail.com>

Repository CRAN

Date/Publication 2026-03-25 22:20:02 UTC

Contents

`mvtelliptical` *Mean and Variance for Truncated Multivariate Elliptical Distributions*

Description

This function approximates the mean vector and variance-covariance matrix for some specific truncated elliptical distributions. The argument `dist` specifies the distribution to be used and accepts the values `"Normal"`, `"t"`, `"PE"`, `"PVII"`, `"Slash"`, and `"CN"`, corresponding to the truncated Normal, Student-t, Power Exponential, Pearson type VII, Slash, and Contaminated Normal distributions, respectively.

Usage

```
mvtelliptical(lower, upper = rep(Inf, length(lower)), mu = rep(0,
  length(lower)), Sigma = diag(length(lower)), dist = "Normal",
  nu = NULL, n = 10000, burn.in = 0, thinning = 3)
```

Arguments

`lower`	vector of lower truncation points of length p.
`upper`	vector of upper truncation points of length p.
`mu`	numeric vector of length p representing the location parameter.
`Sigma`	numeric positive definite matrix with dimension pxp representing the scale parameter.
`dist`	represents the truncated distribution to be used. The values are `'Normal'`, `'t'`, `'PE'`, `'PVII'`, `'Slash'`, and `'CN'` for the truncated Normal, Student-t, Power Exponential, Pearson VII, Slash, and Contaminated Normal distributions, respectively.
`nu`	additional parameter or vector of parameters depending on the density generating function. See Details.
`n`	number of Monte Carlo samples to be generated.
`burn.in`	number of samples to be discarded as a burn-in phase.
`thinning`	factor for reducing the autocorrelation of random points.

Details

Moments associated with the truncated components are estimated using a Monte Carlo approach, while moments for the non-truncated components are obtained by exploiting properties of conditional expectation.

This function also supports the univariate case. The argument `nu` denotes a parameter or a vector of parameters, depending on the underlying density generating function (DGF). For the truncated Student-t, Power Exponential, and Slash distributions, `nu` must be a positive scalar. For the truncated Pearson type VII distribution, `nu` is a vector of length two, where the first element must be greater than $p/2$ and the second element must be strictly positive. For the truncated Contaminated Normal distribution, `nu` is a vector of length two with components taking values in the interval $(0, 1)$.

Value

It returns a list with three elements:

`EY`	the mean vector of length p.
`EYY`	the second moment matrix of dimensions pxp.
`VarY`	the variance-covariance matrix of dimensions pxp.

Note

The Normal distribution is a special case of the Power Exponential distribution obtained when `nu = 1`. The Student-t distribution with ν degrees of freedom arises as a particular case of the Pearson type VII distribution when `nu =` ((ν+p)/2, ν).

For the Student-t distribution, if `nu >= 300`, the Normal approximation is used. The algorithm also supports Student-t distributions with degrees of freedom `nu <= 2`. For the Pearson type VII distribution, the algorithm supports values of `m <= (p + 2)/2`, where `m` corresponds to the first component of `nu`.

Author(s)

Katherine L. Valeriano, Christian E. Galarza and Larissa A. Matos

References

Fang KT, Kotz S, Ng KW (2018). *Symmetric multivariate and related distributions*. Chapman and Hall/CRC.

Galarza CE, Matos LA, Castro LM, Lachos VH (2022). "Moments of the doubly truncated selection elliptical distributions with emphasis on the unified multivariate skew-t distribution." *Journal of Multivariate Analysis*, **189**, 104944. doi:10.1016/j.jmva.2021.104944.

Valeriano KA, Galarza CE, Matos LA (2023). "Moments and random number generation for the truncated elliptical family of distributions." *Statistics and Computing*, **33**(1), 32.

See Also

`rtelliptical`

Examples

```
# Truncated Student-t distribution
set.seed(5678)
mu = c(0.1, 0.2, 0.3)
Sigma = matrix(data = c(1,0.2,0.3,0.2,1,0.4,0.3,0.4,1), nrow=length(mu),
               ncol=length(mu), byrow=TRUE)

# Example 1: considering nu = 0.80 and one doubly truncated variable
a = c(-0.8, -Inf, -Inf)
b = c(0.5, 0.6, Inf)
MC11 = mvtelliptical(a, b, mu, Sigma, "t", 0.80)

# Example 2: considering nu = 0.80 and two doubly truncated variables
a = c(-0.8, -0.70, -Inf)
b = c(0.5, 0.6, Inf)
MC12 = mvtelliptical(a, b, mu, Sigma, "t", 0.80) # By default n=1e4

# Truncated Pearson VII distribution
set.seed(9876)
MC21 = mvtelliptical(a, b, mu, Sigma, "PVII", c(1.90,0.80), n=1e6) # More precision
c(MC12$EY); c(MC21$EY)
MC12$VarY;  MC21$VarY

# Truncated Normal distribution
set.seed(1234)
MC31 = mvtelliptical(a, b, mu, Sigma, "Normal", n=1e4)
MC32 = mvtelliptical(a, b, mu, Sigma, "Normal", n=1e6) # More precision
```

`rtelliptical` *Sampling Random Numbers from Truncated Multivariate Elliptical Distributions*

Description

The sampling procedure is based on a Slice Sampling algorithm combined with Gibbs sampling steps. The distribution is characterized by a location parameter `mu`, a scale matrix `Sigma`, and lower and upper truncation bounds specified by `lower` and `upper`, respectively.

Usage

```
rtelliptical(n = 10000, mu = rep(0, length(lower)),
  Sigma = diag(length(lower)), lower, upper = rep(Inf, length(lower)),
  dist = "Normal", nu = NULL, expr = NULL, gFun = NULL,
  ginvFun = NULL, burn.in = 0, thinning = 1)
```

Arguments

`n`	number of observations to generate. Must be an integer >= 1.
`mu`	numeric vector of length p representing the location parameter.

`Sigma`	numeric positive definite matrix with dimension pxp representing the scale parameter.
`lower`	vector of lower truncation points of length p.
`upper`	vector of upper truncation points of length p.
`dist`	represents the truncated distribution to be used. The values are `'Normal'`, `'t'`, `'PE'`, `'PVII'`, `'Slash'`, and `'CN'` for the truncated Normal, Student-t, Power Exponential, Pearson VII, Slash, and Contaminated Normal distribution, respectively.
`nu`	additional parameter or vector of parameters depending on the density generating function. See Details.
`expr`	a character with the density generating function. See Details.
`gFun`	an R function with the density generating function. See Details.
`ginvFun`	an R function with the inverse of the density generating function defined in `gFun`. See Details.
`burn.in`	number of samples to be discarded as a burn-in phase.
`thinning`	factor for reducing the autocorrelation of random points.

Details

The argument `dist` specifies the truncated distribution to be used and accepts the values `"Normal"`, `"t"`, `"PE"`, `"PVII"`, `"Slash"`, and `"CN"`, corresponding to the truncated Normal, Student-t, Power Exponential, Pearson type VII, Slash, and Contaminated Normal distributions, respectively.

The argument `nu` denotes a parameter or a vector of parameters depending on the underlying density generating function (DGF). For the truncated Student-t, Power Exponential, and Slash distributions, `nu` must be a positive scalar. For the truncated Pearson type VII distribution, `nu` is a vector of length two, where the first element must be greater than $p/2$ and the second element must be strictly positive. For the truncated Contaminated Normal distribution, `nu` is a vector of length two with components taking values in the interval $(0, 1)$.

This function also supports random number generation from truncated elliptical distributions not explicitly listed in the `dist` argument by supplying the density generating function (DGF) through either the `expr` or `gFun` arguments. The DGF must be a non-negative and strictly decreasing function on $(0, \infty)$.

The simplest approach is to provide the DGF to the `expr` argument as a character string. The notation used in `expr` must be compatible with both the **Ryacas** package and the **R** evaluation environment. For example, for the DGF $g(t) = e^{-t}$, the user should specify `expr = "exp(-t)"`. The expression must depend only on the variable t; any additional parameters must be supplied as fixed values. When a character expression is provided via `expr`, the algorithm attempts to compute a closed-form expression for the inverse of $g(t)$. Since such an expression may not always exist, a warning is issued whenever the inversion cannot be obtained analytically (see Example 2).

If random samples cannot be generated using a character expression supplied to `expr`, the user may instead provide a custom **R** function through the `gFun` argument. By default, the inverse of this function is approximated numerically; however, for improved computational efficiency, the user may optionally provide its inverse via the `ginvFun` argument. When `gFun` is supplied, the arguments `dist` and `expr` are ignored.

Value

It returns a matrix of dimensions nxp with the random points sampled.

Note

The Normal distribution is a special case of the Power Exponential distribution obtained when nu = 1. The Student-t distribution with ν degrees of freedom arises as a particular case of the Pearson type VII distribution when nu = $((\nu$+p)/2, $\nu)$.

Author(s)

Katherine L. Valeriano, Christian E. Galarza and Larissa A. Matos

References

Fang KT, Kotz S, Ng KW (2018). *Symmetric multivariate and related distributions*. Chapman and Hall/CRC.

Ho HJ, Lin TI, Chen HY, Wang WL (2012). "Some results on the truncated multivariate t distribution." *Journal of Statistical Planning and Inference*, **142**(1), 25–40. doi:10.1016/j.jspi.2011.06.006.

Neal RM (2003). "Slice sampling." *Annals of statistics*, 705–741.

Robert CP, Casella G (2010). *Introducing Monte Carlo Methods with R*, volume 18. New York: Springer.

Valeriano KA, Galarza CE, Matos LA (2023). "Moments and random number generation for the truncated elliptical family of distributions." *Statistics and Computing*, **33**(1), 32.

See Also

mvtelliptical

Examples

```
library(ggplot2)
library(ggExtra)
library(gridExtra)

# Example 1: Sampling from the Truncated Normal distribution
set.seed(1234)
mu  = c(0, 1)
Sigma = matrix(c(1,0.70,0.70,3), 2, 2)
lower = c(-2, -3)
upper = c(3, 3)
sample1 = rtelliptical(5e4, mu, Sigma, lower, upper, dist="Normal")

# Histogram and density for variable 1
ggplot(data.frame(sample1), aes(x=X1)) +
   geom_histogram(aes(y=..density..), colour="black", fill="grey", bins=15) +
   geom_density(color="red") + labs(x=bquote(X[1]), y="Density") +
   theme_bw()
```

```
# Histogram and density for variable 2
ggplot(data.frame(sample1), aes(x=X2)) +
   geom_histogram(aes(y=..density..), colour="black", fill="grey", bins=15) +
   geom_density(color="red") + labs(x=bquote(X[2]), y="Density") +
   theme_bw()


# Example 2: Sampling from the Truncated Logistic distribution

# Function for plotting the sample autocorrelation using ggplot2
acf.plot = function(samples){
 p = ncol(samples); n = nrow(samples); q1 = qnorm(0.975)/sqrt(n); acf1 = list(p)
 for (i in 1:p){
   bacfdf = with(acf(samples[,i], plot=FALSE), data.frame(lag, acf))
   acf1[[i]] = ggplot(data=bacfdf, aes(x=lag,y=acf)) + geom_hline(aes(yintercept=0)) +
    geom_segment(aes(xend=lag, yend=0)) + labs(x="Lag", y="ACF", subtitle=bquote(X[.(i)])) +
     geom_hline(yintercept=c(q1,-q1), color="red", linetype="twodash") +
     theme_bw()
 }
 return (acf1)
}

set.seed(5678)
mu  = c(0, 0)
Sigma = matrix(c(1,0.70,0.70,1), 2, 2)
lower = c(-2, -2)
upper = c(3, 2)
# Sample autocorrelation with no thinning
sample2 = rtelliptical(2000, mu, Sigma, lower, upper, expr="exp(-t)/(1+exp(-t))^2")
grid.arrange(grobs=acf.plot(sample2), top="Logistic distribution with no thinning", nrow=1)

# Sample autocorrelation with thinning = 3
sample3 = rtelliptical(2000, mu, Sigma, lower, upper, expr="exp(-t)/(1+exp(-t))^2",
                       thinning=3)
grid.arrange(grobs=acf.plot(sample3), top="Logistic distribution with thinning = 3", nrow=1)


# Example 3: Sampling from the Truncated Kotz-type distribution
set.seed(5678)
mu  = c(0, 0)
Sigma = matrix(c(1,-0.5,-0.5,1), 2, 2)
lower = c(-2, -2)
upper = c(3, 2)
sample4 = rtelliptical(2000, mu, Sigma, lower, upper, gFun=function(t){t^(-1/2)*exp(-2*t^(1/4))})
f1 = ggplot(data.frame(sample4), aes(x=X1,y=X2)) + geom_point(size=0.50) +
     labs(x=expression(X[1]), y=expression(X[2]), subtitle="Kotz(2,1/4,1/2)") +
     theme_bw()
ggMarginal(f1, type="histogram", fill="grey")
```

Index