Package ‘relliptical’

January 7, 2022

Type Package

Title The Truncated Elliptical Family of Distributions

Version 1.1.0

Description It offers random numbers generation from members of the truncated multivariate elliptical family of distribution such as the truncated versions of the Normal, Student-t, Pearson VII, Slash, Logistic, among others. Particular distributions can be provided by specifying the density generating function. It also computes the first two moments (covariance matrix as well) for some particular distributions.

License GPL (>= 2)

Encoding UTF-8

Imports FuzzyNumbers.Ext.2, matrixcalc, Rcpp, Rdpack, Ryacas0, stats

RdMacros Rdpack

RoxygenNote 7.1.1

LinkingTo RcppArmadillo, Rcpp

Suggests ggExtra, ggplot2, gridExtra, MomTrunc, TTmoment, tmvtnorm

NeedsCompilation yes

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

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

Repository CRAN

Date/Publication 2022-01-07 13:52:50 UTC

R topics documented:

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` sets the distribution to be used and accepts the same values `Normal`, `t`, `PE`, `PVII`, `Slash`, and `CN`, for the truncated Normal, Student-t, Power Exponential, Pearson VII, Slash, and Contaminated Normal distributions, respectively. Moments are computed through Monte Carlo method for the truncated variables and using properties of the conditional expectation for the non-truncated variables.

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

This function also considers the univariate case. The argument `nu` is a parameter or vector of parameters depending on the density generating function (DGF). For the truncated Student-t, Power Exponential, and Slash distribution, `nu` is a positive number. For the truncated Pearson VII, `nu` is a vector with the first element greater than $p/2$ and the second element a positive number. For the truncated Contaminated Normal distribution, `nu` is a vector of length 2 assuming values between 0 and 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 particular case of the Power Exponential distribution when `nu = 1`. The Student-t distribution with ν degrees of freedom results from the Pearson VII distribution when `nu = ` ((ν+p)/2, ν).

In the Student-t distribution, if `nu >= 300`, the Normal case is considered. For Student-t distribution, the algorithm also supports degrees of freedom `nu <= 2`. For Pearson VII distribution, the algorithm supports values of `m <= (p+2)/2` (first element of `nu`).

Author(s)

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

References

Fang KW (2018). *Symmetric multivariate and related distributions*. CRC Press.

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

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

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

This function generates observations from a truncated multivariate elliptical distribution with location parameter `mu`, scale matrix `Sigma`, lower and upper truncation points `lower` and `upper` via Slice Sampling algorithm with Gibbs sampler steps.

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 distributions, 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 `dist` argument 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.

The argument `nu` is a parameter or vector of parameters depending on the density generating function (DGF). For the truncated Student-t, Power Exponential, and Slash distribution, `nu` is a positive number. For the truncated Pearson VII, `nu` is a vector with the first element greater than $p/2$ and the second element a positive number. For the truncated Contaminated Normal distribution, `nu` is a vector of length 2 assuming values between 0 and 1.

This function also allows generating random numbers from other truncated elliptical distributions not specified in the `dist` argument, by supplying the density generating function (DGF) through arguments either `expr` or `gFun`. The DGF must be a non-negative and strictly decreasing function on `(0,Inf)`. The easiest way is to provide the DGF expression to argument `expr` as a character. The notation used in `expr` needs to be understood by package `Ryacas0`, and the environment of `R`. For instance, for the DGF $g(t) = e^{-t}$, the user must provide `expr = "exp(1)^(-t)"`. See that the function must depend only on variable t, and any additional parameter must be passed as a fixed value. For this case, when a character expression is provided to `expr`, the algorithm tries to compute a closed-form expression for the inverse function of $g(t)$, however, this is not always possible (a warning message is returned). See example 2.

If it was no possible to generate random samples by passing a character expression to `expr`, the user may provide a custom `R` function to the `gFun` argument. By default, its inverse function is approximated numerically, however, the user may also provide its inverse to the `ginvFun` argument to gain some computational time. When `gFun` is provided, 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 particular case of the Power Exponential distribution when `nu = 1`. The Student-t distribution with ν degrees of freedom results from the Pearson VII distribution when `nu = ` ((ν+p)/2, ν).

Author(s)

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

References

Fang KW (2018). *Symmetric multivariate and related distributions*. CRC Press.

Ho HJ, Lin T, Chen H, Wang W (2012). “Some results on the truncated multivariate t distribution.” *Journal of Statistical Planning and Inference*, **142**(1), 25–40.

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

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")

# 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")


# 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")
 }
 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(10000, mu, Sigma, lower, upper, expr="exp(1)^(-t)/(1+exp(1)^(-t))^2")
grid.arrange(grobs=acf.plot(sample2), top="Logistic distribution with no thinning", nrow=1)

# Sample autocorrelation with thinning = 3
sample3 = rtelliptical(10000, mu, Sigma, lower, upper, expr="exp(1)^(-t)/(1+exp(1)^(-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)")
ggMarginal(f1, type="histogram", fill="grey")
```

Index