

Package ‘tmvtnorm’

October 7, 2009

Version 0.8-3

Date 2009-10-06

Title Truncated Multivariate Normal Distribution

Author Stefan Wilhelm <wilhelm@financial.com>

Maintainer Stefan Wilhelm <wilhelm@financial.com>

Imports stats, graphics

Depends R (>= 1.9.0), mvtnorm, utils

Description Computes truncated multivariate normal probabilities, quantiles and densities, including one-dimensional and bivariate marginal densities. Random number generation. Computes first and second moments (i.e. mean and covariance matrix) for the double-truncated case.

License GPL (>= 2)

URL <http://www.r-project.org>

Repository CRAN

Date/Publication 2009-10-07 07:48:53

R topics documented:

dtmvnorm-marginal	2
dtmvnorm.marginal2	4
mtmvnorm	6
ptmvnorm	8
ptmvnorm.marginal	9
qtmvnorm-marginal	10
rtmvnorm	12
tmvnorm	15

Index	18
--------------	-----------

dtmvnorm-marginal *One-dimensional marginal density functions from a Truncated Multivariate Normal distribution*

Description

This function computes the one-dimensional marginal density function from a Truncated Multivariate Normal density function using the algorithm given in Cartinhour (1990).

Usage

```
dtmvnorm.marginal(xn, n=1,
  mean= rep(0, nrow(sigma)),
  sigma=diag(length(mean)),
  lower=rep(-Inf, length = length(mean)),
  upper=rep( Inf, length = length(mean)))
```

Arguments

xn	Vector of quantiles to calculate the marginal density for.
n	Index position (1..k) within the random vector x to calculate the one-dimensional marginal density for.
mean	Mean vector, default is rep(0, length = nrow(sigma)).
sigma	Covariance matrix, default is diag(length(mean)).
lower	Vector of lower truncation points,\ default is rep(-Inf, length = length(mean)).
upper	Vector of upper truncation points,\ default is rep(Inf, length = length(mean)).

Details

The one-dimensional marginal density $f_i(x_i)$ of x_i is

$$f_i(x_i) = \int_{a_1}^{b_1} \dots \int_{a_{i-1}}^{b_{i-1}} \int_{a_{i+1}}^{b_{i+1}} \dots \int_{a_k}^{b_k} f(x) dx_{-i}$$

Note that the one-dimensional marginal density is not truncated normal, but only conditional densities are truncated normal.

Author(s)

Stefan Wilhelm <Stefan.Wilhelm@financial.com>

References

Cartinhour, J. (1990). One-dimensional marginal density functions of a truncated multivariate normal density function. *Communications in Statistics - Theory and Methods*, **19**, 197–203

Arnold et al. (1993). The Nontruncated Marginal of a Truncated Bivariate Normal Distribution. *Psychometrika*, **58**, 471–488

Examples

```
#####
#
# Example 1: truncated bivariate normal
#
#####

# parameters of the bivariate normal distribution
sigma = matrix(c(1, 0.95,
                 0.95, 1), 2, 2)
mu = c(0,0)

# sample from multivariate normal distribution
X = rmvnorm(5000, mu, sigma)

# tuncation in x2 with x2 <= 0
X.trunc = X[X[,2]<0,]

# plot the realisations before and after truncation
par(mfrow=c(2,2))
plot(X, col="gray", xlab=expression(x[1]), ylab=expression(x[2]),
     main="realisations from a\n truncated bivariate normal distribution")
points(X.trunc)
abline(h=0, lty=2, col="gray")
#legend("topleft", col=c("gray", "black"))

# marginal density for x1 from realisations
plot(density(X.trunc[,1]), main=expression("marginal density for "*x[1]))

# one-dimensional marginal density for x1 using the formula
x <- seq(-5, 5, by=0.01)
fx <- dtmvnorm.marginal(x, n=1, mean=mu, sigma=sigma,
  lower=c(-Inf,-Inf), upper=c(Inf,0))
lines(x, fx, lwd=2, col="red")

# marginal density for x2
plot(density(X.trunc[,2]), main=expression("marginal density for "*x[2]))

# one-dimensional marginal density for x2 using the formula
x <- seq(-5, 5, by=0.01)
fx <- dtmvnorm.marginal(x, n=2, mean=mu, sigma=sigma,
  lower=c(-Inf,-Inf), upper=c(Inf,0))
lines(x, fx, lwd=2, col="blue")

#####
#
# Example 2 : truncated trivariate normal
#
#####

# parameters of the trivariate normal distribution
sigma = outer(1:3,1:3,pmin)
```

```

mu      = c(0,0,0)

# sample from multivariate normal distribution
X       = rmvnorm(2000, mu, sigma)

# truncation in x2 and x3 : x2 <= 0, x3 <= 0
X.trunc = X[X[,2]<=0 & X[,3]<=0,]

par(mfrow=c(2,3))
plot(X, col="gray", xlab=expression(x[1]), ylab=expression(x[2]),
     main="realisations from a\n truncated trivariate normal distribution")
points(X.trunc, col="black")
abline(h=0, lty=2, col="gray")

plot(X[,2:3], col="gray", xlab=expression(x[2]), ylab=expression(x[3]),
     main="realisations from a\n truncated trivariate normal distribution")
points(X.trunc[,2:3], col="black")
abline(h=0, lty=2, col="gray")
abline(v=0, lty=2, col="gray")

plot(X[,c(1,3)], col="gray", xlab=expression(x[1]), ylab=expression(x[3]),
     main="realisations from a\n truncated trivariate normal distribution")
points(X.trunc[,c(1,3)], col="black")
abline(h=0, lty=2, col="gray")

# one-dimensional marginal density for x1 from realisations and formula
plot(density(X.trunc[,1]), main=expression("marginal density for "*x[1])),
x <- seq(-5, 5, by=0.01)
fx <- dtmvnorm.marginal(x, n=1, mean=mu, sigma=sigma,
  lower=c(-Inf,-Inf,-Inf), upper=c(Inf,0,0))
lines(x, fx, lwd=2, col="red")

# one-dimensional marginal density for x2 from realisations and formula
plot(density(X.trunc[,2]), main=expression("marginal density for "*x[2])),
x <- seq(-5, 5, by=0.01)
fx <- dtmvnorm.marginal(x, n=2, mean=mu, sigma=sigma,
  lower=c(-Inf,-Inf,-Inf), upper=c(Inf,0,0))
lines(x, fx, lwd=2, col="red")

# one-dimensional marginal density for x3 from realisations and formula
plot(density(X.trunc[,3]), main=expression("marginal density for "*x[3])),
x <- seq(-5, 5, by=0.01)
fx <- dtmvnorm.marginal(x, n=3, mean=mu, sigma=sigma,
  lower=c(-Inf,-Inf,-Inf), upper=c(Inf,0,0))
lines(x, fx, lwd=2, col="red")

```

Description

This function computes the bivariate marginal density function $f(x_q, x_r)$ from a k-dimensional Truncated Multivariate Normal density function ($k \geq 2$). The bivariate marginal density is obtained by integrating out $(k-2)$ dimensions as proposed by Tallis (1961). This function is basically an extraction of the Leppard and Tallis (1989) Fortran code for moments calculation, but extended to the double truncated case.

Usage

```
dtmvnorm.marginal2(xq, xr, q, r,
  mean = rep(0, nrow(sigma)),
  sigma = diag(length(mean)),
  lower = rep(-Inf, length = length(mean)),
  upper = rep(Inf, length = length(mean)))
```

Arguments

xq	Value x_q
xr	Value x_r
q	Index position for x_q within mean vector to calculate the bivariate marginal density for.
r	Index position for x_r within mean vector to calculate the bivariate marginal density for.
mean	Mean vector, default is <code>rep(0, length = nrow(sigma))</code> .
sigma	Covariance matrix, default is <code>diag(length(mean))</code> .
lower	Vector of lower truncation points, default is <code>rep(-Inf, length = length(mean))</code> .
upper	Vector of upper truncation points, default is <code>rep(Inf, length = length(mean))</code> .

Details

The bivariate marginal density function $f(x_q, x_r)$ for $x \sim TN(\mu, \Sigma, a, b)$ and $q \neq r$ is defined as

$$F_{q,r}(x_q = c_q, x_r = c_r) = \int_{a_1}^{b_1} \dots \int_{a_{q-1}}^{b_{q-1}} \int_{a_{q+1}}^{b_{q+1}} \dots \int_{a_{r-1}}^{b_{r-1}} \int_{a_{r+1}}^{b_{r+1}} \dots \int_{a_k}^{b_k} \varphi_{\alpha\Sigma}(x_s, c_q, c_r) dx_s$$

Author(s)

Stefan Wilhelm <Stefan.Wilhelm@financial.com>, Manjunath B G <bgmanjunath@gmail.com>

References

- Tallis, G. M. (1961). The moment generating function of the truncated multinormal distribution. *Journal of the Royal Statistical Society, Series B*, **23**, 223–229
- Leppard, P. and Tallis, G. M. (1989). Evaluation of the Mean and Covariance of the Truncated Multinormal *Applied Statistics*, **38**, 543–553
- Manjunath B G and Wilhelm, S. (2009). Moments Calculation For the Double Truncated Multivariate Normal Density. Working Paper. Available at SSRN: <http://ssrn.com/abstract=1472153>

Examples

```

lower = c(-0.5, -1, -1)
upper = c( 2.2,  2,  2)

mean = c(0,0,0)
sigma = matrix(c(2.0, -0.6,  0.7,
                 -0.6,  1.0, -0.2,
                 0.7, -0.2,  1.0), 3, 3)

# generate random samples from untruncated and truncated distribution
Y = rmvnorm(10000, mean=mean, sigma=sigma)
X = rtmvnorm(500, mean=mean, sigma=sigma, lower=lower, upper=upper,
            algorithm="gibbs")

# compute bivariate marginal density of x1 and x2
xq <- seq(lower[1], upper[1], by=0.1)
xr <- seq(lower[2], upper[2], by=0.1)

grid <- matrix(NA, length(xq), length(xr))
for (i in 1:length(xq))
{
  for (j in 1:length(xr))
  {
    grid[i,j] = dtmvnorm.marginal2(xq=xq[i], xr=xr[j],
                                   q=1, r=2, sigma=sigma, lower=lower, upper=upper)
  }
}

plot(Y[,1], Y[,2], xlim=c(-4, 4), ylim=c(-4, 4),
      main=expression("bivariate marginal density (*x[1]*", "*x[2]*")"),
      xlab=expression(x[1]), ylab=expression(x[2]), col="gray80")
points(X[,1], X[,2], col="black")

lines(x=c(lower[1], upper[1], upper[1], lower[1], lower[1]),
      y=c(lower[2], lower[2], upper[2], upper[2], lower[2]),
      lty=2, col="red")
contour(xq, xr, grid, add=TRUE, nlevels = 8, col="red", lwd=2)

# scatterplot matrices for untruncated and truncated points
require(lattice)
splom(Y)
splom(X)

```

Description

Computation of the first two moments, i.e. mean vector and covariance matrix for the Truncated Multivariate Normal Distribution based on the works of Tallis (1961), Lee (1979) and Leppard and Tallis (1989), but extended to the double-truncated case with general mean and general covariance matrix.

Usage

```
mtmvnorm(mean, sigma, lower, upper)
```

Arguments

mean	Mean vector, default is <code>rep(0, length = ncol(x))</code> .
sigma	Covariance matrix, default is <code>diag(ncol(x))</code> .
lower	Vector of lower truncation points,\ default is <code>rep(-Inf, length = length(mean))</code> .
upper	Vector of upper truncation points,\ default is <code>rep(Inf, length = length(mean))</code> .

Value

mean	Mean vector of truncated variables
cov	Covariance matrix of truncated variables

Author(s)

Stefan Wilhelm <Stefan.Wilhelm@financial.com>, Manjunath B G <bgmanjunath@gmail.com>

References

Tallis, G. M. (1961). The moment generating function of the truncated multinormal distribution. *Journal of the Royal Statistical Society, Series B*, **23**, 223–229

Lee, L.-F. (1979). On the first and second moments of the truncated multi-normal distribution and a simple estimator. *Economics Letters*, **3**, 165–169

Leppard, P. and Tallis, G. M. (1989). Evaluation of the Mean and Covariance of the Truncated Multinormal. *Applied Statistics*, **38**, 543–553

Manjunath B G and Wilhelm, S. (2009). Moments Calculation For the Double Truncated Multivariate Normal Density. Working Paper. Available at SSRN: <http://ssrn.com/abstract=1472153>

Examples

```
mu <- c(0.5, 0.5, 0.5)
sigma <- matrix(c( 1, 0.6, 0.3,
                 0.6, 1, 0.2,
                 0.3, 0.2, 2), 3, 3)

a <- c(-Inf, -Inf, -Inf)
b <- c(1, 1, 1)
```

```
# compute first and second moments
mtmvnorm(mu, sigma, lower=a, upper=b)

# compare with simulated results
X <- rtmvnorm(n=1000, mean=mu, sigma=sigma, lower=a, upper=b)
colMeans(X)
cov(X)
```

ptmvnorm

Truncated Multivariate Normal Distribution

Description

Computes the distribution function of the truncated multivariate normal distribution for arbitrary limits and correlation matrices based on the `pmvnorm()` implementation of the algorithms by Genz and Bretz.

Usage

```
ptmvnorm(lowerx, upperx, mean=rep(0, length(lowerx)), sigma,
  lower = rep(-Inf, length = length(mean)),
  upper = rep( Inf, length = length(mean)),
  maxpts = 25000, abseps = 0.001, releps = 0)
```

Arguments

<code>lowerx</code>	the vector of lower limits of length <code>n</code> .
<code>upperx</code>	the vector of upper limits of length <code>n</code> .
<code>mean</code>	the mean vector of length <code>n</code> .
<code>sigma</code>	the covariance matrix of dimension <code>n</code> . Either <code>corr</code> or <code>sigma</code> can be specified. If <code>sigma</code> is given, the problem is standardized. If neither <code>corr</code> nor <code>sigma</code> is given, the identity matrix is used for <code>sigma</code> .
<code>lower</code>	Vector of lower truncation points,\ default is <code>rep(-Inf, length = length(mean))</code> .
<code>upper</code>	Vector of upper truncation points,\ default is <code>rep(Inf, length = length(mean))</code> .
<code>maxpts</code>	maximum number of function values as integer.
<code>abseps</code>	absolute error tolerance as double.
<code>releps</code>	relative error tolerance as double.

Details

The computation of truncated multivariate normal probabilities and densities is done using conditional probabilities from the standard/untruncated multivariate normal distribution. So we refer to the documentation of the `mvtnorm` package and the methodology is described in Genz (1992, 1993).

Value

The evaluated distribution function is returned with attributes

error estimated absolute error and
msg status messages.

References

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, **1**, 141–150

Genz, A. (1993). Comparison of methods for the computation of multivariate normal probabilities. *Computing Science and Statistics*, **25**, 400–405

Johnson, N./Kotz, S. (1970). Distributions in Statistics: Continuous Multivariate Distributions Wiley & Sons, pp. 70–73

Horrace, W. (2005). Some Results on the Multivariate Truncated Normal Distribution. *Journal of Multivariate Analysis*, **94**, 209–221

ptmvnorm.marginal *One-dimensional marginal CDF function from a Truncated Multivariate Normal distribution*

Description

This function computes the one-dimensional marginal probability function from a Truncated Multivariate Normal density function using integration in `pmvnorm()`.

Usage

```
ptmvnorm.marginal(xn,
  n = 1,
  mean = rep(0, nrow(sigma)),
  sigma = diag(length(mean)),
  lower = rep(-Inf, length = length(mean)),
  upper = rep(Inf, length = length(mean)))
```

Arguments

xn	Vector of quantiles to calculate the marginal probability for.
n	Index position (1..k) within the random vector xn to calculate the one-dimensional marginal probability for.
mean	the mean vector of length k.
sigma	the covariance matrix of dimension k. Either <code>corr</code> or <code>sigma</code> can be specified. If <code>sigma</code> is given, the problem is standardized. If neither <code>corr</code> nor <code>sigma</code> is given, the identity matrix is used for <code>sigma</code> .
lower	Vector of lower truncation points,\ default is <code>rep(-Inf, length = length(mean))</code> .
upper	Vector of upper truncation points,\ default is <code>rep(Inf, length = length(mean))</code> .

Details

The one-dimensional marginal probability for index i is $F_i(x_i) = P(X_i \leq x_i)$

$$F_i(x_i) = \int_{a_1}^{b_1} \dots \int_{a_{i-1}}^{b_{i-1}} \int_{a_i}^{x_i} \int_{a_{i+1}}^{b_{i+1}} \dots \int_{a_k}^{b_k} f(x) dx = \alpha^{-1} \Phi_k(a, u, \mu, \Sigma)$$

where $u = (b_1, \dots, b_{i-1}, x_i, b_{i+1}, \dots, b_k)'$ is the upper integration bound and Φ_k is the k -dimensional normal probability (i.e. function `pmvnorm()` in R package `mvtnorm`).

Value

Returns a vector of the same length as `xn` with probabilities.

Author(s)

Stefan Wilhelm <Stefan.Wilhelm@financial.com>

Examples

```
##
lower=c(-1,-1,-1)
upper=c(1,1,1)
mean=c(0,0,0)
sigma=matrix(c(1, 0.8, 0.2,
              0.8, 1, 0.1,
              0.2, 0.1, 1), 3, 3)

X = rtmvnorm(n=1000, mean=c(0,0,0), sigma=sigma, lower=lower, upper=upper)

x = seq(-1, 1, by=0.001)
Fx = ptmvnorm.marginal(xn=x, n=1, mean=c(0,0,0), sigma=sigma, lower=lower, upper=upper)

plot(ecdf(X[,1]))
lines(x, Fx, type="l", col="blue")
```

qtmvnorm-marginal *Quantiles of the Truncated Multivariate Normal Distribution in one dimension*

Description

Computes the equicoordinate quantile function of the truncated multivariate normal distribution for arbitrary correlation matrices based on an inversion of the algorithms by Genz and Bretz.

Usage

```
qtmvnorm.marginal(p,
  interval = c(-10, 10),
  tail = c("lower.tail", "upper.tail", "both.tails"),
  n=1,
  mean=rep(0, nrow(sigma)),
  sigma=diag(length(mean)),
  lower=rep(-Inf, length = length(mean)),
  upper=rep( Inf, length = length(mean)),
  ...)
```

Arguments

p	probability.
interval	a vector containing the end-points of the interval to be searched by uniroot .
tail	specifies which quantiles should be computed. <code>lower.tail</code> gives the quantile x for which $P[X \leq x] = p$, <code>upper.tail</code> gives x with $P[X > x] = p$ and <code>both.tails</code> leads to x with $P[-x \leq X \leq x] = p$.
n	index (1..n) to calculate marginal quantile for
mean	the mean vector of length n.
sigma	the covariance matrix of dimension n. Either <code>corr</code> or <code>sigma</code> can be specified. If <code>sigma</code> is given, the problem is standardized. If neither <code>corr</code> nor <code>sigma</code> is given, the identity matrix is used for <code>sigma</code> .
lower	Vector of lower truncation points,\ default is <code>rep(-Inf, length = length(mean))</code> .
upper	Vector of upper truncation points,\ default is <code>rep(Inf, length = length(mean))</code> .
...	additional parameters to be passed to uniroot .

Details

Only equicoordinate quantiles are computed, i.e., the quantiles in each dimension coincide. Currently, the distribution function is inverted by using the [uniroot](#) function which may result in limited accuracy of the quantiles.

Value

A list with four components: `quantile` and `f.quantile` give the location of the quantile and the value of the function evaluated at that point. `iter` and `estim.prec` give the number of iterations used and an approximate estimated precision from [uniroot](#).

See Also

[ptmvnorm](#), [pmvnorm](#)

Examples

```

# finite dimensional distribution of the Geometric Brownian Motion log-returns
# with truncation

# volatility p.a.
sigma=0.4

# risk free rate
r = 0.05

# n=3 points in time
T <- c(0.5, 0.7, 1)

# covariance matrix of Geometric Brownian Motion returns
Sigma = sigma^2*outer(T,T,pmin)

# mean vector of the Geometric Brownian Motion returns
mu    = (r - sigma^2/2) * T

# lower truncation vector a (a<=x<=b)
a = rep(-Inf, 3)

# upper truncation vector b (a<=x<=b)
b = c(0, 0, Inf)

# quantile of the t_1 returns
qtmvnorm.marginal(p=0.95, interval = c(-10, 10), tail = "lower.tail", n=1,
  mean = mu, sigma = Sigma, lower=a, upper=b)

```

rtmvnorm

*Sampling Random Numbers From Truncated Multivariate Normal
Distribution*

Description

This function generates random numbers from the truncated multivariate normal distribution with mean equal to mean and covariance matrix sigma, lower and upper truncation points lower and upper with either rejection sampling or Gibbs sampling.

Usage

```

rtmvnorm(n, mean = rep(0, nrow(sigma)),
  sigma = diag(length(mean)),
  lower=rep(-Inf, length = length(mean)),
  upper=rep( Inf, length = length(mean)),
  algorithm=c("rejection", "gibbs", "gibbsR"),
  ...)

```

Arguments

n	Number of observations.
mean	Mean vector, default is <code>rep(0, length = ncol(x))</code> .
sigma	Covariance matrix, default is <code>diag(ncol(x))</code> .
lower	Vector of lower truncation points, default is <code>rep(-Inf, length = length(mean))</code> .
upper	Vector of upper truncation points, default is <code>rep(Inf, length = length(mean))</code> .
algorithm	Method used, possible methods are rejection sampling ("rejection", default), the Fortan Gibbs sampler ("gibbs") and the old Gibbs sampler implementation in R ("gibbsR").
...	additional parameters for Gibbs sampling, given to <code>rmvtnorm.gibbs</code> , such as <code>burn.in.samples</code> and <code>start.value</code>

Details

The generation of random numbers from a truncated multivariate normal distribution is done using either rejection sampling or Gibbs sampler.

Rejection sampling is done from the standard multivariate normal distribution. So we use the function `rmvnorm` of the `mvtnorm` package. In order to speed up the generation of N samples from the truncated distribution, we first calculate the acceptance rate alpha from the truncation points and then generate N/alpha samples iteratively until we have got N samples. This typically does not take more than 2-3 iterations. Rejection sampling may be very inefficient when the support region is small (i.e. in higher dimensions) which results in very low acceptance rates alpha. In this case the Gibbs sampler is preferable.

The Gibbs sampler samples from univariate conditional distributions, so all samples can be accepted except for a burn-in period. The number of burn-in samples to be discarded can be specified, as well as a start value. If no start value is given we determine a start value from the support region using either lower bound or upper bound if they are finite, or 0 otherwise.

The Gibbs sampler has been reimplemented in Fortran 90 for performance reasons (`algorithm="gibbs"`). The old R implementation is still accessible through `algorithm="gibbsR"`.

Author(s)

Stefan Wilhelm <Stefan.Wilhelm@financial.com>, Manjunath B G <bgmanjunath@gmail.com>

References

- Johnson, N./Kotz, S. (1970). Distributions in Statistics: Continuous Multivariate Distributions *Wiley & Sons*, pp. 70–73
- Horrace, W. (2005). Some Results on the Multivariate Truncated Normal Distribution. *Journal of Multivariate Analysis*, **94**, 209–221
- Jayesh H. Kotecha and Petar M. Djuric (1999). Gibbs Sampling Approach For Generation of Truncated Multivariate Gaussian Random Variables *IEEE Computer Society*, 1757–1760

See Also

[ptmvnorm](#), [pmvnorm](#), [rmvnorm](#), [dmvnorm](#)

Examples

```

dtmvnorm(x=c(0,0))
dtmvnorm(x=c(0,0), mean=c(1,1), upper=c(0,0))

#####
#
# Example 1:
# rejection sampling
#
#####

sigma <- matrix(c(4,2,2,3), ncol=2)
x <- rtmvnorm(n=500, mean=c(1,2), sigma=sigma, upper=c(1,0))
plot(x, main="samples from truncated bivariate normal distribution",
      xlim=c(-6,6), ylim=c(-6,6),
      xlab=expression(x[1]), ylab=expression(x[2]))
abline(v=1, lty=3, lwd=2, col="gray")
abline(h=0, lty=3, lwd=2, col="gray")

#####
#
# Example 2:
# Gibbs sampler for 4 dimensions
#
#####

C = matrix(0.8, 4, 4)
diag(C)=rep(1, 4)
lower = rep(-4, 4)
upper = rep(-1, 4)

# acceptance rate alpha
alpha = pmvnorm(lower=lower, upper=upper, mean=rep(0,4), sigma=C)
alpha

# Gibbs sampler
X1=rtmvnorm(n=20000, mean = rep(0,4), sigma=C, lower=lower, upper=upper,
            algorithm="gibbs", burn.in.samples=100)
# Rejection sampling
X2=rtmvnorm(n=5000, mean = rep(0,4), sigma=C, lower=lower, upper=upper)

colMeans(X1)
colMeans(X2)

plot(density(X1[,1]), col="red", lwd=2, main="Gibbs vs. Rejection")
lines(density(X2[,1]), col="blue", lwd=2)
legend("topleft", legend=c("Gibbs Sampling", "Rejection Sampling"),
      col=c("red", "blue"), lwd=2)

```

tmvnorm

*Truncated Multivariate Normal Density***Description**

These functions provide the density function for the truncated multivariate normal distribution with mean equal to `mean` and covariance matrix `sigma`, lower and upper truncation points `lower` and `upper`

Usage

```
dtmvnorm(x, mean = rep(0, nrow(sigma)),
         sigma = diag(length(mean)),
         lower=rep(-Inf, length = length(mean)),
         upper=rep( Inf, length = length(mean)),
         log=FALSE)
```

Arguments

<code>x</code>	Vector or matrix of quantiles. If <code>x</code> is a matrix, each row is taken to be a quantile.
<code>mean</code>	Mean vector, default is <code>rep(0, length = ncol(x))</code> .
<code>sigma</code>	Covariance matrix, default is <code>diag(ncol(x))</code> .
<code>lower</code>	Vector of lower truncation points,\ default is <code>rep(-Inf, length = length(mean))</code> .
<code>upper</code>	Vector of upper truncation points,\ default is <code>rep(Inf, length = length(mean))</code> .
<code>log</code>	Logical; if TRUE, densities <code>d</code> are given as <code>log(d)</code> .

Details

The computation of truncated multivariate normal probabilities and densities is done using conditional probabilities from the standard/untruncated multivariate normal distribution. So we refer to the documentation of the `mvtnorm` package and the methodology is described in Genz (1992, 1993).

Author(s)

Stefan Wilhelm <Stefan.Wilhelm@financial.com>

References

- Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, **1**, 141–150
- Genz, A. (1993). Comparison of methods for the computation of multivariate normal probabilities. *Computing Science and Statistics*, **25**, 400–405
- Johnson, N./Kotz, S. (1970). *Distributions in Statistics: Continuous Multivariate Distributions* Wiley & Sons, pp. 70–73
- Horrace, W. (2005). Some Results on the Multivariate Truncated Normal Distribution. *Journal of Multivariate Analysis*, **94**, 209–221

See Also

[ptmvnorm](#), [pmvnorm](#), [rmvnorm](#), [dmvnorm](#)

Examples

```

dtmvnorm(x=c(0,0))
dtmvnorm(x=c(0,0), mean=c(1,1), upper=c(0,0))

#####
#
# Example 1:
# truncated multivariate normal density
#
#####

x1<-seq(-2, 3, by=0.1)
x2<-seq(-2, 3, by=0.1)

density<-function(x)
{
  sigma=matrix(c(1, -0.5, -0.5, 1), 2, 2)
  z=dtmvnorm(x, mean=c(0,0), sigma=sigma, lower=c(-1,-1))
  z
}

fgrid <- function(x, y, f)
{
  z <- matrix(nrow=length(x), ncol=length(y))
  for(m in 1:length(x)){
    for(n in 1:length(y)){
      z[m,n] <- f(c(x[m], y[n]))
    }
  }
  z
}

# compute density d for grid
d=fgrid(x1, x2, density)

# plot density as contourplot
contour(x1, x2, d, nlevels=5, main="Truncated Multivariate Normal Density",
  xlab=expression(x[1]), ylab=expression(x[2]))
abline(v=-1, lty=3, lwd=2)
abline(h=-1, lty=3, lwd=2)

#####
#
# Example 2:
# generation of random numbers
# from a truncated multivariate normal distribution
#
#####

```

```
sigma <- matrix(c(4,2,2,3), ncol=2)
x <- rtmvnorm(n=500, mean=c(1,2), sigma=sigma, upper=c(1,0))
plot(x, main="samples from truncated bivariate normal distribution",
      xlim=c(-6,6), ylim=c(-6,6),
      xlab=expression(x[1]), ylab=expression(x[2]))
abline(v=1, lty=3, lwd=2, col="gray")
abline(h=0, lty=3, lwd=2, col="gray")
```

Index

*Topic **distribution**

dtmnorm-marginal, 2
dtmnorm.marginal2, 4
mtmnorm, 6
ptmnorm, 8
ptmnorm.marginal, 9
qtmnorm-marginal, 10
rtmnorm, 12
tmvnorm, 15

*Topic **multivariate**

dtmnorm-marginal, 2
dtmnorm.marginal2, 4
mtmnorm, 6
ptmnorm, 8
ptmnorm.marginal, 9
qtmnorm-marginal, 10
rtmnorm, 12
tmvnorm, 15

dmvnorm, 13, 16
dtmnorm (*tmvnorm*), 15
dtmnorm-marginal, 2
dtmnorm.marginal
 (*dtmnorm-marginal*), 2
dtmnorm.marginal2, 4

moments (*mtmnorm*), 6
mtmnorm, 6

pmvnorm, 11, 13, 16
ptmnorm, 8, 11, 13, 16
ptmnorm.marginal, 9

qtmnorm-marginal, 10
qtmnorm.marginal
 (*qtmnorm-marginal*), 10

rmvnorm, 13, 16
rtmnorm, 12

tmvnorm, 15

uniroot, 11