Package ‘condTruncMVN’

October 12, 2022

Version 0.0.2

Type Package

Title Conditional Truncated Multivariate Normal Distribution

Maintainer Paul M. Hargarten <hargartenp@vcu.edu>

Description Computes the density and probability for the conditional truncated multivariate normal (Horrace (2005) p. 4, <doi:10.1016/j.jmva.2004.10.007>). Also draws random samples from this distribution.

License GPL-3

Depends R(>= 3.5.0)

Imports condMVNorm(>= 2020.1), matrixNormal(>= 0.0.1), tmvmixnorm(>= 1.0.2), tmvtnorm(>= 1.4-10), truncnorm(>= 1.0-8)

Suggests formatR, knitr, rmarkdown, roxygen2, sessioninfo, spelling, testthat

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Language en-US

BugReports https://github.com/phargarten2/condTruncMVN/issues

VignetteBuilder knitr

NeedsCompilation no

Author Paul M. Hargarten [aut, cre]

Repository CRAN

Date/Publication 2020-09-17 12:50:11 UTC

R topics documented:

---

condtMVN *Conditional Truncated Multivariate Normal Parameters*

Description

Suppose that **Z = (X,Y)** is from a fully-joint multivariate normal distribution of dimension *n* with `mean` and covariance matrix `sigma` truncated between `lower` and `upper`. This function provides the parameters for the conditional mean and covariance matrix of **Y** given **X**. See the vignette for more information.

Usage

```
condtMVN(
  mean,
  sigma,
  lower,
  upper,
  dependent.ind,
  given.ind,
  X.given,
  init = rep(0, length(mean))
)
```

Arguments

`mean`	the mean vector for Z of length of n
`sigma`	the symmetric and positive-definite covariance matrix of dimension n x n of Z.
`lower`	a vector of lower bounds of length n that truncate Z
`upper`	a vector of upper bounds of length n that truncate Z
`dependent.ind`	a vector of integers denoting the indices of dependent variable Y.
`given.ind`	a vector of integers denoting the indices of conditioning variable X. If specified as integer vector of length zero or left unspecified, the unconditional density is returned.
`X.given`	a vector of reals denoting the conditioning value of X. This should be of the same length as `given.ind`
`init`	initial value used for random generation of truncated multivariate normal in a Gibbs sampler. Default: A vector of zeros, equal to the number of components. For details, see tmvmixnorm::`rtmvn`().

Details

The first four arguments are the parameters of multivariate normal and the truncation space. `dependent.ind`, `given.ind`, `X.given`, `init` are all arguments that determines the conditional truncated MVN.

Using the full data **Z**, the conditional mean and conditional variance of **Y|X** are determined (Wang, 2006). Additionally, to reflect the reduced dimension of **Y|X**, the truncation limits are also adjusted.

See the vignette for more information.

Value

Returns a list of:

- condMean - conditional mean of **Y|X**
- condVar - conditional variance of **Y|X**
- condLower - the lower bound of **Y|X**
- condUpper - the upper bound of **Y|X**
- condInit - the initial values adjusted to match the dimension of **Y|X**. These are used to randomly generate the truncated multivariate normal rcmvtruncnorm.

Note

This function is based on `condMVN` from the **condMVNorm** package.

References

Wang, R. 2006. Appendix A: Marginal and conditional distributions of multivariate normal distribution. `http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node7.html`.

See Also

`cmvnorm`, `pmvnorm`, `Mvnorm`

Examples

```
# Suppose X2,X3,X5|X2,X4 ~ N_3(1, Sigma) and truncated between -10 and 10.
d <- 5
rho <- 0.9
Sigma <- matrix(0, nrow = d, ncol = d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

# Conditional Truncated Normal Parameters
condtMVN(mean = rep(1, d),
  sigma = Sigma,
  lower = rep(-10, d),
  upper = rep(10, d),
  dependent.ind = c(2, 3, 5),
  given.ind = c(1, 4), X.given = c(1, -1)
)
```

`dcmvtruncnorm` *Density of the Conditional Truncated Multivariate Normal*

Description

Calculates the density of truncated conditional multivariate normal Y|X: $f(Y = y|X = X.given)$. See the vignette for more information.

Usage

```
dcmvtruncnorm(
  y,
  mean,
  sigma,
  lower,
  upper,
  dependent.ind,
  given.ind,
  X.given,
  log = FALSE
)
```

Arguments

`y`	vector or matrix of quantiles of Y. If a matrix, each row is taken to be a quantile. This is the quantity that the density is calculated from.
`mean`	the mean vector for Z of length of n
`sigma`	the symmetric and positive-definite covariance matrix of dimension n x n of Z.
`lower`	a vector of lower bounds of length n that truncate Z
`upper`	a vector of upper bounds of length n that truncate Z
`dependent.ind`	a vector of integers denoting the indices of dependent variable Y.
`given.ind`	a vector of integers denoting the indices of conditioning variable X. If specified as integer vector of length zero or left unspecified, the unconditional density is returned.
`X.given`	a vector of reals denoting the conditioning value of X. This should be of the same length as `given.ind`
`log`	logical; if `TRUE`, densities d are given as log(d).

References

Horrace, W.C. 2005. Some results on the multivariate truncated normal distribution. Journal of Multivariate Analysis, 94, 209–221. https://surface.syr.edu/cgi/viewcontent.cgi?article=1149&context=ecn

Examples

```
# Example 1: X2,X3,X5|X2,X4 ~ N_3(1, Sigma)
# truncated between -10 and 10.
d <- 5
rho <- 0.9
Sigma <- matrix(0, nrow = d, ncol = d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

# Log-density of 0
dcmvtruncnorm(
  rep(0, 3),
  mean = rep(1, 5),
  sigma = Sigma,
  lower = rep(-10, 5),
  upper = rep(10, d),
  dependent.ind = c(2, 3, 5),
  given.ind = c(1, 4), X.given = c(1, -1),
  log = TRUE
)
```

---

pcmvtruncnorm — *CDF for the Conditional Truncated Multivariate Normal*

Description

Computes the distribution function for a conditional truncated multivariate normal random variate **Y**|**X**.

Usage

```
pcmvtruncnorm(
  lowerY,
  upperY,
  mean,
  sigma,
  lower,
  upper,
  dependent.ind,
  given.ind,
  X.given,
  ...
)
```

Arguments

`lowerY`	the vector of lower limits for **Y**|**X**. Passed to tmvtnorm::`ptmvnorm`().
`upperY`	the vector of upper limits for **Y**|**X**. Must be greater than lowerY. Passed to tmvtnorm::`ptmvnorm`().

`mean`	the mean vector for Z of length of n
`sigma`	the symmetric and positive-definite covariance matrix of dimension n x n of Z.
`lower`	a vector of lower bounds of length n that truncate Z
`upper`	a vector of upper bounds of length n that truncate Z
`dependent.ind`	a vector of integers denoting the indices of dependent variable Y.
`given.ind`	a vector of integers denoting the indices of conditioning variable X. If specified as integer vector of length zero or left unspecified, the unconditional density is returned.
`X.given`	a vector of reals denoting the conditioning value of X. This should be of the same length as `given.ind`
`...`	Additional arguments passed to tmvtnorm::`ptmvnorm`(). The CDF is calculated using the Genz algorithm based on these arguments: maxpts, abseps, and releps.

Details

Calculates the probability that **Y|X** is between `lowerY` and `upperY`. Z = (X, Y) is the fully joint multivariate normal distribution with mean equal mean and covariance matrix sigma, truncated between lower and upper. See the vignette for more information.

Note

For one-dimension conditionals **Y|X**, this function uses the ptruncnorm() function in the truncnorm package. Otherwise, this function uses tmvtnorm::`ptmvnorm`().

Examples

```
# Example 1: Let X2,X3,X5|X2,X4 ~ N_3(1, Sigma)
# truncated between -10 and 10.
d <- 5
rho <- 0.9
Sigma <- matrix(0, nrow = d, ncol = d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

# Find P(-0.5 < X2,X3,X5 < 0 | X2,X4)
pcmvtruncnorm(rep(-0.5, 3), rep(0, 3),
  mean = rep(1, d),
  sigma = Sigma,
  lower = rep(-10, d),
  upper = rep(10, d),
  dependent.ind = c(2, 3, 5),
  given.ind = c(1, 4), X.given = c(1, -1)
)

# Example 2: Let X1| X2 = 1, X3 = -1, X4 = 1, X5 = -1 ~ N(1, Sigma) truncated
# between -10 and 10. Find P(-0.5 < X1 < 0 | X2 = 1, X3 = -1, X4 = 1, X5 = -1).
pcmvtruncnorm(-0.5, 0,
  mean = rep(1, d),
  sigma = Sigma,
  lower = rep(-10, d),
```

```
  upper = rep(10, d),
  dependent.ind = 1,
  given.ind = 2:5, X.given = c(1, -1, 1, -1)
)
```

---

rcmvtruncnorm *Random Sample from Conditional Truncated Multivariate Normal*

Description

Randomly samples from conditional truncated multivariate normal distribution variate, **Y|X**, where **Z = (X, Y)** is the fully joint multivariate normal distribution with `mean`, covariance matrix `sigma`, and truncated between `lower` and `upper`. See the vignette for more information.

Usage

```
rcmvtruncnorm(
  n,
  mean,
  sigma,
  lower,
  upper,
  dependent.ind,
  given.ind,
  X.given,
  init = rep(0, length(mean)),
  burn = 10L,
  thin = 1
)
```

Arguments

`n`	number of random samples desired (sample size).
`mean`	the mean vector for Z of length of n
`sigma`	the symmetric and positive-definite covariance matrix of dimension n x n of Z.
`lower`	a vector of lower bounds of length n that truncate Z
`upper`	a vector of upper bounds of length n that truncate Z
`dependent.ind`	a vector of integers denoting the indices of dependent variable Y.
`given.ind`	a vector of integers denoting the indices of conditioning variable X. If specified as integer vector of length zero or left unspecified, the unconditional density is returned.
`X.given`	a vector of reals denoting the conditioning value of X. This should be of the same length as `given.ind`
`init`	initial value used for random generation of truncated multivariate normal in a Gibbs sampler. Default: A vector of zeros, equal to the number of components. For details, see tmvmixnorm::`rtmvn`().

burn	the burn-in, which is the number of initial iterations to be discarded. Default: 10. Passed to rtmvn().
thin	thinning lag (default as 1).

Note

Uses rtmvn from the **tmvmixnorm** package to find the random variate.

Examples

```
# Generate 2 random numbers from X2,X3,X5|X2,X4 ~ N_3(1, Sigma)
# truncated between -10 and 10.
d <- 5
rho <- 0.9
Sigma <- matrix(0, nrow = d, ncol = d)
Sigma <- rho^abs(row(Sigma) - col(Sigma))

set.seed(2342)
rcmvtruncnorm(2,
  mean = rep(1, d),
  sigma = Sigma,
  lower = rep(-10, d),
  upper = rep(10, d),
  dependent.ind = c(2, 3, 5),
  given.ind = c(1, 4), X.given = c(1, -1)
)

# Example 2: Generate two random numbers from
# X1|X2, X3, X4, X5 ~ N(1, Sigma) truncated between -10 and 10.
set.seed(2342)
rcmvtruncnorm(2,
  mean = rep(1, d),
  sigma = Sigma,
  lower = rep(-10, d),
  upper = rep(10, d),
  dependent.ind = 1,
  given.ind = 2:5, X.given = c(1, -1, 1, -1)
)
```

Index