Package ‘mvtnorm’

June 9, 2020

Title Multivariate Normal and t Distributions
Version 1.1-1
Date 2020-06-09
Description Computes multivariate normal and t probabilities, quantiles,
random deviates and densities.
Imports stats, methods
Depends R(>= 3.5.0)
License GPL-2
URL http://mvtnorm.R-forge.R-project.org
NeedsCompilation yes
Author Alan Genz [aut],
Frank Bretz [aut],
Tetsuhisa Miwa [aut],
Xuefei Mi [aut],
Friedrich Leisch [ctb],
Fabian Scheipl [ctb],
Bjoern Bornkamp [ctb] (<https://orcid.org/0000-0002-6294-8185>),
Martin Maechler [ctb],
Torsten Hothorn [aut, cre] (<https://orcid.org/0000-0001-8301-0471>)
Maintainer Torsten Hothorn <Torsten.Hothorn@R-project.org>
Repository CRAN
Date/Publication 2020-06-09 15:50:02 UTC

R topics documented:

  algorithms ................................................................. 2
  Mvnorm ................................................................. 3
  Mvt ................................................................. 4
  pmvnorm ............................................................... 6
  pmvt ............................................................... 9
  qmvnorm ............................................................. 12
  qmvt ............................................................. 14
Choice of Algorithm and Hyper Parameters

Description
Choose between three algorithms for evaluating normal distributions and define hyper parameters.

Usage
GenzBretz(maxpts = 25000, abseps = 0.001, releps = 0)
Miwa(steps = 128)
TVPACK(abseps = 1e-6)

Arguments
maxpts maximum number of function values as integer. The internal FORTRAN code always uses a minimum number depending on the dimension. (for example 752 for three-dimensional problems).
abseps absolute error tolerance; for TVPACK only used for dimension 3.
releps relative error tolerance as double.
steps number of grid points to be evaluated.

Details
There are three algorithms available for evaluating normal probabilities: The default is the randomized Quasi-Monte-Carlo procedure by Genz (1992, 1993) and Genz and Bretz (2002) applicable to arbitrary covariance structures and dimensions up to 1000.

For smaller dimensions (up to 20) and non-singular covariance matrices, the algorithm by Miwa et al. (2003) can be used as well.

For two- and three-dimensional problems and semi-infinite integration region, TVPACK implements an interface to the methods described by Genz (2004).

Value
An object of class GenzBretz or Miwa defining hyper parameters.

References
**Mvnorm**


---

**Mvnorm**

*Multivariate Normal Density and Random Deviates*

**Description**

These functions provide the density function and a random number generator for the multivariate normal distribution with mean equal to `mean` and covariance matrix `sigma`.

**Usage**

```r
dmvnorm(x, mean = rep(0, p), sigma = diag(p), log = FALSE, checkSymmetry = TRUE)
rmvnorm(n, mean = rep(0, nrow(sigma)), sigma = diag(length(mean)),
       method=c("eigen", "svd", "chol"), pre0.9_9994 = FALSE, checkSymmetry = TRUE)
```

**Arguments**

- `x`: vector or matrix of quantiles. If `x` is a matrix, each row is taken to be a quantile.
- `n`: number of observations.
- `mean`: mean vector, default is `rep(0,length = ncol(x))`.
- `sigma`: covariance matrix, default is `diag(ncol(x))`.
- `log`: logical; if TRUE, densities `d` are given as `log(d)`.
- `method`: string specifying the matrix decomposition used to determine the matrix root of `sigma`. Possible methods are eigenvalue decomposition ("eigen", default), singular value decomposition ("svd"), and Cholesky decomposition ("chol"). The Cholesky is typically fastest, not by much though.
- `pre0.9_9994`: logical; if FALSE, the output produced in mvtnorm versions up to 0.9-9993 is reproduced. In 0.9-9994, the output is organized such that `rmvnorm(10,...)` has the same first ten rows as `rmvnorm(100,...)` when called with the same seed.
- `checkSymmetry`: logical; if FALSE, skip checking whether the covariance matrix is symmetric or not. This will speed up the computation but may cause unexpected outputs when ill-behaved `sigma` is provided. The default value is TRUE.
The Multivariate t Distribution

These functions provide information about the multivariate t distribution with non-centrality parameter (or mode) delta, scale matrix sigma and degrees of freedom df. dmvt gives the density and rmvt generates random deviates.

**Usage**

```r
rmvt(n, sigma = diag(2), df = 1, delta = rep(0, nrow(sigma)),
     type = c("shifted", "Kshirsagar"), ...)
dmvt(x, delta = rep(0, p), sigma = diag(p), df = 1, log = TRUE,
     type = "shifted", checkSymmetry = TRUE)
```

**Arguments**

- **x** vector or matrix of quantiles. If `x` is a matrix, each row is taken to be a quantile.
- **n** number of observations.
- **delta** the vector of noncentrality parameters of length `n`, for type = "shifted" delta specifies the mode.
- **sigma** scale matrix, defaults to diag(ncol(x)).
If $X$ denotes a random vector following a $t$ distribution with location vector $0$ and scale matrix $\Sigma$ (written $X \sim t_\nu(0, \Sigma)$), the scale matrix (the argument sigma) is not equal to the covariance matrix $\text{Cov}(X)$ of $X$. If the degrees of freedom $\nu$ (the argument df) is larger than 2, then $\text{Cov}(X) = \Sigma\nu/(\nu - 2)$. Furthermore, in this case the correlation matrix $\text{Cor}(X)$ equals the correlation matrix corresponding to the scale matrix $\Sigma$ (which can be computed with \text{cov2cor}()). Note that the scale matrix is sometimes referred to as “dispersion matrix”; see McNeil, Frey, Embrechts (2005, p. 74).

For type = "shifted" the density

$$c(1 + (x - \delta)^\prime S^{-1}(x - \delta)/\nu)^{-(\nu + m)/2}$$

is implemented, where

$$c = \Gamma((\nu + m)/2)/((\pi\nu)^{m/2}\Gamma(\nu/2)|S|^{1/2}),$$

$S$ is a positive definite symmetric matrix (the matrix sigma above), $\delta$ is the non-centrality vector and $\nu$ are the degrees of freedom.

df=0 historically leads to the multivariate normal distribution. From a mathematical point of view, rather df=Inf corresponds to the multivariate normal distribution. This is (now) also allowed for \text{rmvt}() and \text{dmvt}().

Note that \text{dmvt}() has default log = TRUE, whereas \text{dmvnorm}() has default log = FALSE.

References

See Also

`pmvt()` and `qmvt()`

Examples

```r
## basic evaluation
dmvt(x = c(0,0), sigma = diag(2))

## check behavior for df=0 and df=Inf
x <- c(1.23, 4.56)
mu <- 1:2
Sigma <- diag(2)
x0 <- dmvt(x, delta = mu, sigma = Sigma, df = 0) # default log = TRUE!
x8 <- dmvt(x, delta = mu, sigma = Sigma, df = Inf) # default log = TRUE!	xn <- dmvnorm(x, mean = mu, sigma = Sigma, log = TRUE)
stopifnot(identical(x0, x8), identical(x0, xn))

## X ~ t_3(0, diag(2))
x <- rmvt(100, sigma = diag(2), df = 3) # t_3(0, diag(2)) sample
plot(x)

## X ~ t_3(mu, Sigma)
n <- 1000
mu <- 1:2
Sigma <- matrix(c(4, 2, 2, 3), ncol=2)
set.seed(271)
x <- rep(mu, each=n) + rmvt(n, sigma=Sigma, df=3)
plot(x)

## Note that the call rmvt(n, mean=mu, sigma=Sigma, df=3) does *not*
## give a valid sample from t_3(mu, Sigma)! [and thus throws an error]
try(rmvt(n, mean=mu, sigma=Sigma, df=3))

## df=Inf correctly samples from a multivariate normal distribution
set.seed(271)
x <- rep(mu, each=n) + rmvt(n, sigma=Sigma, df=Inf)
set.seed(271)
x. <- rmvnorm(n, mean=mu, sigma=Sigma)
stopifnot(identical(x, x.))
```

---

**pmvnorm**

*Multivariate Normal Distribution*

**Description**

Computes the distribution function of the multivariate normal distribution for arbitrary limits and correlation matrices.
pmvnorm

Usage

pmvnorm(lower=-Inf, upper=Inf, mean=rep(0, length(lower)),
       corr=NULL, sigma=NULL, algorithm = GenzBretz(), ...)  

Arguments

lower the vector of lower limits of length n.
upper the vector of upper limits of length n.
mean the mean vector of length n.
corr the correlation matrix of dimension n.
sigma the covariance matrix of dimension n less than 1000. Either corr or sigma can be specified. If sigma is given, the problem is standardized. If neither corr nor sigma is given, the identity matrix is used for sigma.
algorithm an object of class GenzBretz, Miwa or TVPACK specifying both the algorithm to be used as well as the associated hyper parameters.
... additional parameters (currently given to GenzBretz for backward compatibility issues).

Details

This program involves the computation of multivariate normal probabilities with arbitrary correlation matrices. It involves both the computation of singular and nonsingular probabilities. The implemented methodology is described in Genz (1992, 1993) (for algorithm GenzBretz), in Miwa et al. (2003) for algorithm Miwa (useful up to dimension 20) and Genz (2004) for the TVPACK algorithm (which covers 2- and 3-dimensional problems for semi-infinite integration regions).

Note the default algorithm GenzBretz is randomized and hence slightly depends on Random.seed and that both -Inf and +Inf may be specified in lower and upper. For more details see pmvt.

The multivariate normal case is treated as a special case of pmvt with df=0 and univariate problems are passed to pnorm.

The multivariate normal density and random deviates are available using dmvnorm and rmvnorm.

Value

The evaluated distribution function is returned with attributes

error estimated absolute error and
msg status messages.

Source

http://www.sci.wsu.edu/math/faculty/genz/homepage
References


See Also

`qmvnorm`

Examples

```r
n <- 5
generate_random(mean <- rep(0, 5), lower <- rep(-1, 5), upper <- rep(3, 5), corr <- diag(5))
corr[lower.tri(corr)] <- 0.5
corr[upper.tri(corr)] <- 0.5
prob <- pmvnorm(lower, upper, mean, corr)
print(prob)

stopifnot(pmvnorm(lower=-Inf, upper=3, mean=0, sigma=1) == pnorm(3))

a <- pmvnorm(lower=-Inf,upper=c(.3,.5),mean=c(2,4),diag(2))
stopifnot(round(a,16) == round(prod(pnorm(c(.3,.5),c(2,4))),16))

a <- pmvnorm(lower=-Inf,upper=c(.3,.5,1),mean=c(2,4,1),diag(3))
stopifnot(round(a,16) == round(prod(pnorm(c(.3,.5,1),c(2,4,1))),16))

# Example from R News paper (original by Genz, 1992):

m <- 3
sigma <- diag(3)
sigma[2,1] <- 3/5
sigma[3,1] <- 1/3
sigma[3,2] <- 11/15
pmvnorm(lower=rep(-Inf, m), upper=c(1,4,2), mean=rep(0, m), corr=sigma)

# Correlation and Covariance
```
```r
a <- pmvnorm(lower=-Inf, upper=c(2,2), sigma = diag(2)*2)
b <- pmvnorm(lower=-Inf, upper=c(2,2)/sqrt(2), corr=diag(2))
stopifnot(all.equal(round(a,5), round(b, 5)))
```

---

**pmvt**

* Multivariate t Distribution

**Description**

Computes the distribution function of the multivariate t distribution for arbitrary limits, degrees of freedom and correlation matrices based on algorithms by Genz and Bretz.

**Usage**

```r
pmvt(lower=-Inf, upper=Inf, delta=rep(0, length(lower)),
df=1, corr=NULL, sigma=NULL, algorithm = GenzBretz(),
type = c("Kshirsagar", "shifted"), ...)
```

**Arguments**

- `lower` the vector of lower limits of length n.
- `upper` the vector of upper limits of length n.
- `delta` the vector of noncentrality parameters of length n, for type = "shifted" delta specifies the mode.
- `df` degree of freedom as integer. Normal probabilities are computed for df=0.
- `corr` the correlation matrix of dimension n.
- `sigma` the scale matrix of dimension n. Either corr or sigma can be specified. If sigma is given, the problem is standardized. If neither corr nor sigma is given, the identity matrix is used for sigma.
- `algorithm` an object of class GenzBretz or TVPACK defining the hyper parameters of this algorithm.
- `type` type of the noncentral multivariate t distribution to be computed. type = "Kshirsagar" corresponds to formula (1.4) in Genz and Bretz (2009) (see also Chapter 5.1 in Kotz and Nadarajah (2004)). This is the noncentral t-distribution needed for calculating the power of multiple contrast tests under a normality assumption. type = "shifted" corresponds to the formula right before formula (1.4) in Genz and Bretz (2009) (see also formula (1.1) in Kotz and Nadarajah (2004)). It is a location shifted version of the central t-distribution. This noncentral multivariate t distribution appears for example as the Bayesian posterior distribution for the regression coefficients in a linear regression. In the central case both types coincide.
- `...` additional parameters (currently given to GenzBretz for backward compatibility issues).
Details

This function involves the computation of central and noncentral multivariate t-probabilities with arbitrary correlation matrices. It involves both the computation of singular and nonsingular probabilities. The methodology (for default algorithm = GenzBretz()) is based on randomized quasi Monte Carlo methods and described in Genz and Bretz (1999, 2002).

Because of the randomization, the result for this algorithm (slightly) depends on .Random.seed.

For 2- and 3-dimensional problems one can also use the TVPACK routines described by Genz (2004), which only handles semi-infinite integration regions (and for type = "Kshirsagar" only central problems).

For type = "Kshirsagar" and a given correlation matrix corr, for short A, say, (which has to be positive semi-definite) and degrees of freedom $\nu$ the following values are numerically evaluated

$$I = 2^{1-\nu/2}/\Gamma(\nu/2) \int_0^{\infty} s^{\nu-1} \exp(-s^2/2) \Phi(s \cdot lower/\sqrt{\nu - \delta}, s \cdot upper/\sqrt{\nu - \delta}) ds$$

where

$$\Phi(a, b) = (\text{det}(A)(2\pi)^{m})^{-1/2} \int_a^b \exp(-x'Ax/2) dx$$

is the multivariate normal distribution and $m$ is the number of rows of $A$.

For type = "shifted", a positive definite symmetric matrix $S$ (which might be the correlation or the scale matrix), mode (vector) $\delta$ and degrees of freedom $\nu$ the following integral is evaluated:

$$c \int_{lower_1}^{upper_1} \cdots \int_{lower_m}^{upper_m} (1 + (x - \delta)'S^{-1}(x - \delta)/\nu)^{-\nu+m)/2} dx_1...dx_m,$$

where

$$c = \Gamma((\nu + m)/2)/(\pi^{m/2}\Gamma(\nu/2)|S|^{1/2}),$$

and $m$ is the number of rows of $S$.

Note that both -Inf and +Inf may be specified in the lower and upper integral limits in order to compute one-sided probabilities.

Univariate problems are passed to pt. If df = 0, normal probabilities are returned.

Value

The evaluated distribution function is returned with attributes

- error: estimated absolute error and
- msg: status message (a character string).

Source

http://www.sci.wsu.edu/math/faculty/genz/homepage
pmvt

References


See Also

qmvt

Examples

```r
n <- 5
lower <- -1
upper <- 3
df <- 4
corr <- diag(5)
corr[lower.tri(corr)] <- 0.5
delta <- rep(0, 5)
prob <- pmvt(lower=lower, upper=upper, delta=delta, df=df, corr=corr)
print(prob)

pmvt(lower=-Inf, upper=3, df = 3, sigma = 1) == pt(3, 3)

# Example from R News paper (original by Edwards and Berry, 1987)

n <- c(26, 24, 20, 33, 32)
V <- diag(1/n)
df <- 130
C <- c(1,1,0,0,-1,0,0,0,0,-1,0,0,0,0,-1,-1,0,0,-1,0,0,0)
C <- matrix(C, ncol=5)
### scale matrix
cv <- C %*% V %*% t(C)
### correlation matrix
dv <- t(1/sqrt(diag(cv)))
CR <- cv * (t(dv) %*% dv)
delta <- rep(0,5)

myfct <- function(q, alpha) {
```
```
lower <- rep(-q, ncol(cv))
upper <- rep(q, ncol(cv))
pmvt(lower=lower, upper=upper, delta=delta, df=df, 
corr=cr, abseps=0.0001) - alpha
}

### uniroot for this simple problem
round(uniroot(myfct, lower=1, upper=5, alpha=0.95)$root, 3)

# compare pmvt and pmvnorm for large df:
a <- pmvnorm(lower=-Inf, upper=1, mean=rep(0, 5), corr=diag(5))
b <- pmvt(lower=-Inf, upper=1, delta=rep(0, 5), df=300, 
corr=diag(5))
a b

stopifnot(round(a, 2) == round(b, 2))

# correlation and scale matrix
a <- pmvt(lower=-Inf, upper=2, delta=rep(0,5), df=3, 
sigma = diag(5)*2)
b <- pmvt(lower=-Inf, upper=2/sqrt(2), delta=rep(0,5), 
df=3, corr=diag(5))
attributes(a) <- NULL
attributes(b) <- NULL
a b

stopifnot(all.equal(round(a,3) , round(b, 3)))

a <- pmvt(0, 1, df=10)
attributes(a) <- NULL
b <- pt(1, df=10) - pt(0, df=10)
stopifnot(all.equal(round(a,10) , round(b, 10)))
```

---

### qmvnorm

**Quantiles of the Multivariate Normal Distribution**

**Description**

Computes the equicoordinate quantile function of the multivariate normal distribution for arbitrary correlation matrices based on inversion of `pmvnorm`, using a stochastic root finding algorithm described in Bornkamp (2018).

**Usage**

```
qmvnorm(p, interval = NULL, tail = c("lower.tail", 
  "upper.tail", "both.tails"), mean = 0, corr = NULL,
```
qmvnorm

```r
sigma = NULL, algorithm = GenzBretz(),
ptol = 0.001, maxiter = 500, trace = FALSE, ...)
```

**Arguments**

- `p` probability.
- `interval` optional, a vector containing the end-points of the interval to be searched. Does not need to contain the true quantile, just used as starting values by the root-finder. If equal to NULL a guess is used.
- `tail` specifies which quantiles should be computed. `lower.tail` gives the quantile \( x \) for which \( P[X \leq x] = p \), `upper.tail` gives \( x \) with \( P[X > x] = p \) and `both.tails` leads to \( x \) with \( P[-x \leq X \leq x] = p \).
- `mean` the mean vector of length \( n \).
- `corr` the correlation matrix of dimension \( n \).
- `sigma` the covariance matrix of dimension \( n \). Either `corr` or `sigma` can be specified. If `sigma` is given, the problem is standardized. If neither `corr` nor `sigma` is given, the identity matrix is used for `sigma`.
- `algorithm` an object of class `GenzBretz`, `Miwa` or `TVPACK` specifying both the algorithm to be used as well as the associated hyper parameters.
- `ptol`, `maxiter`, `trace`
  - Parameters passed to the stochastic root-finding algorithm. Iteration stops when the 95% confidence interval for the predicted quantile is inside \( [p - ptol, p + ptol] \).
  - `maxiter` is the maximum number of iterations for the root finding algorithm.
  - `trace` prints the iterations of the root finder.
- `...`
  - additional parameters to be passed to `GenzBretz`.

**Details**

Only equicoordinate quantiles are computed, i.e., the quantiles in each dimension coincide. The result is seed dependend.

**Value**

A list with two components: `quantile` and `f.quantile` give the location of the quantile and the difference between the distribution function evaluated at the quantile and \( p \).

**References**


**See Also**

- `pmvnorm`, `qmvnorm`

**Examples**

```r
qmvnorm(0.95, sigma = diag(2), tail = "both")
```
Description

Computes the equicoordinate quantile function of the multivariate $t$ distribution for arbitrary correlation matrices based on inversion of $pmvt$, using a stochastic root finding algorithm described in Bornkamp (2018).

Usage

$qmvt(p, \text{interval} = \text{NULL}, \text{tail} = \text{c("lower.tail", 
"upper.tail", "both.tails")}, \text{df} = 1, \text{delta} = 0, \text{corr} = \text{NULL}, 
\text{sigma} = \text{NULL}, \text{algorithm} = \text{GenzBretz()}, 
\text{type} = \text{c("Kshirsagar", "shifted")}, 
\text{ptol} = 0.001, \text{maxiter} = 500, \text{trace} = \text{FALSE}, ...)$

Arguments

- $p$: probability.
- $\text{interval}$: optional, a vector containing the end-points of the interval to be searched. Does not need to contain the true quantile, just used as starting values by the root-finder. If equal to NULL a guess is used.
- $\text{tail}$: specifies which quantiles should be computed. lower.tail gives the quantile $x$ for which $P[X \leq x] = p$, upper.tail gives $x$ with $P[X > x] = p$ and both.tails leads to $x$ with $P[-x \leq X \leq x] = p$.
- $\text{delta}$: the vector of noncentrality parameters of length $n$, for type = "shifted" delta specifies the mode.
- $\text{df}$: degree of freedom as integer. Normal quantiles are computed for $df = 0$ or $df = \text{Inf}$.
- $\text{corr}$: the correlation matrix of dimension $n$.
- $\text{sigma}$: the covariance matrix of dimension $n$. Either corr or sigma can be specified. If sigma is given, the problem is standardized. If neither corr nor sigma is given, the identity matrix in the univariate case (so corr = 1) is used for corr.
- $\text{algorithm}$: an object of class GenzBretz or TVPACK defining the hyper parameters of this algorithm.
- $\text{type}$: type of the noncentral multivariate $t$ distribution to be computed. type = "Kshirsagar" corresponds to formula (1.4) in Genz and Bretz (2009) (see also Chapter 5.1 in Kotz and Nadarajah (2004)) and type = "shifted" corresponds to the formula before formula (1.4) in Genz and Bretz (2009) (see also formula (1.1) in Kotz and Nadarajah (2004)).
- $\text{ptol}$, $\text{maxiter}$, $\text{trace}$: Parameters passed to the stochastic root-finding algorithm. Iteration stops when the 95% confidence interval for the predicted quantile is inside $[p-\text{ptol}, p+\text{ptol}]$. maxiter is the maximum number of iterations for the root finding algorithm. trace prints the iterations of the root finder.
additional parameters to be passed to \texttt{GenzBretz}.

**Details**

Only equicoordinate quantiles are computed, i.e., the quantiles in each dimension coincide. The result is seed dependend.

**Value**

A list with two components: \texttt{quantile} and \texttt{f.quantile} give the location of the quantile and the difference between the distribution function evaluated at the quantile and \( p \).

**References**


**See Also**

\texttt{pmvnorm, qmvnorm}

**Examples**

```r
## basic evaluation
qmvt(0.95, df = 16, tail = "both")

## check behavior for df=0 and df=Inf
Sigma <- diag(2)
set.seed(29)
q0 <- qmvt(0.95, sigma = Sigma, df = 0, tail = "both")$quantile
set.seed(29)
q8 <- qmvt(0.95, sigma = Sigma, df = Inf, tail = "both")$quantile
set.seed(29)
qn <- qmvnorm(0.95, sigma = Sigma, tail = "both")$quantile
stopifnot(identical(q0, q8),
           isTRUE(all.equal(q0, qn, tol = (.Machine$double.eps)^((1/3))))

## if neither sigma nor corr are provided, corr = 1 is used internally
df <- 0
set.seed(29)
qt95 <- qmvt(0.95, df = df, tail = "both")$quantile
set.seed(29)
qt95.c <- qmvt(0.95, df = df, corr = 1, tail = "both")$quantile
set.seed(29)
qt95.s <- qmvt(0.95, df = df, sigma = 1, tail = "both")$quantile
stopifnot(identical(qt95, qt95.c),
           identical(qt95, qt95.s))

df <- 4
set.seed(29)
qt95 <- qmvt(0.95, df = df, tail = "both")$quantile
set.seed(29)
```
qt95.c <- qmvt(0.95, df = df, corr = 1, tail = "both")$quantile
set.seed(29)
qt95.s <- qmvt(0.95, df = df, sigma = 1, tail = "both")$quantile
stopifnot(identical(qt95, qt95.c),
          identical(qt95, qt95.s))
Index

*Topic **distribution**
  algorithms, 2
  Mvnorm, 3
  Mvt, 4
  pmvnorm, 6
  pmvt, 9
  qmvnorm, 12
  qmvt, 14

*Topic **multivariate**
  Mvnorm, 3
  Mvt, 4
  .Random.seed, 7, 10
  algorithms, 2
  character, 10
  cov2cor, 5
  
  dmvnorm, 5, 7
  dmvnorm (Mvnorm), 3
  dmv (Mvt), 4

  GenzBretz, 7, 9, 13–15
  GenzBretz (algorithms), 2

  logical, 5

  Miwa, 7, 13
  Miwa (algorithms), 2
  Mvnorm, 3
  Mvt, 4
  
  pmvnorm, 4, 6, 12, 13, 15
  pmvt, 5–7, 9, 14
  pnorm, 7
  pt, 10

  qmvnorm, 4, 8, 12, 15
  qmvt, 6, 11, 13, 14
  
  rmvnorm, 5, 7