Package ‘mniw’

August 22, 2022

Type Package
Title The Matrix-Normal Inverse-Wishart Distribution
Version 1.0.1
Date 2022-08-11
Description Density evaluation and random number generation for the Matrix-Normal Inverse-Wishart (MNIW) distribution, as well as the the Matrix-Normal, Matrix-T, Wishart, and Inverse-Wishart distributions. Core calculations are implemented in a portable (header-only) C++ library, with matrix manipulations using the 'Eigen' library for linear algebra. Also provided is a Gibbs sampler for Bayesian inference on a random-effects model with multivariate normal observations.

URL https://github.com/mlysy/mniw/

BugReports https://github.com/mlysy/mniw/issues
License GPL-3
Depends R (>= 2.10)
Imports Rcpp (>= 0.11.6)
LinkingTo Rcpp, RcppEigen
LazyData true
Suggests testthat, knitr, rmarkdown
Encoding UTF-8
RoxygenNote 7.2.1
VignetteBuilder knitr
NeedsCompilation yes
Author Martin Lysy [aut, cre], Bryan Yates [aut]
Maintainer Martin Lysy <mlysy@uwaterloo.ca>
Repository CRAN
Date/Publication 2022-08-22 16:30:05 UTC
## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>mniw-package</td>
<td>2</td>
</tr>
<tr>
<td>crossprodV</td>
<td>3</td>
</tr>
<tr>
<td>Hospitals</td>
<td>3</td>
</tr>
<tr>
<td>MatrixNormal</td>
<td>4</td>
</tr>
<tr>
<td>MatrixT</td>
<td>5</td>
</tr>
<tr>
<td>MNIW</td>
<td>6</td>
</tr>
<tr>
<td>MultiNormal</td>
<td>7</td>
</tr>
<tr>
<td>rRxNorm</td>
<td>8</td>
</tr>
<tr>
<td>RxNormLM</td>
<td>9</td>
</tr>
<tr>
<td>Wishart</td>
<td>13</td>
</tr>
</tbody>
</table>

### Description

Density evaluation and random number generation for the Matrix-Normal Inverse-Wishart (MNIW) distribution, as well as its constituent distributions, i.e., the Matrix-Normal, Matrix-T, Wishart, and Inverse-Wishart distributions.

### Details

The Matrix-Normal Inverse-Wishart (MNIW) distribution \((X, V) \sim \text{MNIW}(\Lambda, \Sigma, \Psi, \nu)\) on random matrices \(X_{p \times q}\) and symmetric positive-definite \(V_{q \times q}\) is defined as

\[
V \sim \text{Inverse-Wishart}(\Psi, \nu)
\]

\[
X \mid V \sim \text{Matrix-Normal}(\Lambda, \Sigma, V),
\]

where the Matrix-Normal distribution is defined as the multivariate normal

\[
\text{vec}(X) \sim \mathcal{N}(\text{vec}(\Lambda), V \otimes \Sigma),
\]

where \(\text{vec}(X)\) is a vector stacking the columns of \(X\), and \(V \otimes \Sigma\) denotes the Kronecker product.

### Author(s)

**Maintainer:** Martin Lysy <mlysy@uwaterloo.ca>

Authors:

- Bryan Yates

### See Also

Useful links:

- [https://github.com/mlysy/mniw/](https://github.com/mlysy/mniw/)
crossprodV

Matrix cross-product.

Description

Vectorized matrix cross-products $t(X) V Y$ or $t(X) V^{-1} Y$.

Usage

crossprodV(X, Y = NULL, V, inverse = FALSE)

Arguments

X 
A matrix of size $p \times q$, or an array of size $p \times q \times n$.

Y 
A matrix of size $p \times r$, or an array of size $p \times r \times n$. If missing defaults to $Y = X$.

V 
A matrix of size $p \times p$, or an array of size $p \times p \times n$.

inverse 
Logical; whether or not the inner product should be calculated with $V$ or $V^{-1}$.

Value

An array of size $q \times r \times n$.

Examples

# problem dimensions
p <- 4
q <- 2
r <- 3
n <- 5
X <- array(rnorm(p*q*n), dim = c(p, q, n)) # vectorized
Y <- array(rnorm(p*r*n), dim = c(p, r, n)) # vectorized
V <- crossprod(matrix(rnorm(p*p), p, p)) # not vectorized (but positive definite)
crossprodV(X = X, V = V) # self cross-product
# cross-product with inverse matrix weight
crossprodV(X = X, V = V, Y = Y, inverse = TRUE)

Hospitals

Hospital profiling data.

Description

Information on patient-reported problem rates for 27 teaching hospitals and private academic health centers in the United States.
Usage
Hospitals

Format
A data frame with 27 rows (one for each hospital) and 4 variables:

- **NSrg**: Non-surgery related problem rate (%).
- **Srg**: Surgery related problem rate (%).
- **Severity**: Average health index for surveyed patients.
- **Size**: Number of patients surveyed.

References

---


**Description**
Density and random sampling for the Matrix-Normal distribution.

**Usage**
- `dMNorm(X, Lambda, SigmaR, SigmaC, log = FALSE)`
- `rMNorm(n, Lambda, SigmaR, SigmaC)`

**Arguments**
- **X**: Argument to the density function. Either a \( p \times q \) matrix or a \( p \times q \times n \) array.
- **Lambda**: Mean parameter. Either a \( p \times q \) matrix or a \( p \times q \times n \) array.
- **SigmaR**: Between-row covariance matrix. Either a \( p \times p \) matrix or a \( p \times p \times n \) array.
- **SigmaC**: Between-column covariance matrix. Either a \( q \times q \) matrix or a \( q \times q \times n \) array.
- **log**: Logical; whether or not to compute the log-density.
- **n**: Integer number of random samples to generate.

**Details**
The Matrix-Normal distribution \( X \sim \text{Matrix-Normal}(\Lambda, \Sigma_R, \Sigma_C) \) on the random matrix \( X_{p \times q} \) is defined as
\[
\text{vec}(X) \sim \mathcal{N}(\text{vec}(\Lambda), \Sigma_C \otimes \Sigma_R),
\]
where \( \text{vec}(X) \) is a vector stacking the columns of \( X \), and \( \Sigma_C \otimes \Sigma_R \) denotes the Kronecker product.
MatrixT

Value

A vector length \( n \) for density evaluation, or an array of size \( p \times q \times n \) for random sampling.

Examples

```r
# problem dimensions
p <- 4
d <- 2
n <- 10 # number of observations

# parameter values
Lambda <- matrix(rnorm(p*q), p, q) # mean matrix
SigmaR <- crossprod(matrix(rnorm(p*p), p, p)) # row-wise variance matrix (positive definite)
SigmaC <- rwish(n, Psi = diag(q), nu = q + 1) # column-wise variance (vectorized)

# random sample
X <- rMNorm(n, Lambda = Lambda, SigmaR = SigmaR, SigmaC = SigmaC)

# log-density at each sampled value
dMNorm(X, Lambda = Lambda, SigmaR = SigmaR, SigmaC = SigmaC, log = TRUE)
```

MatrixT

The Matrix-t distribution.

Description

Density and sampling for the Matrix-t distribution.

Usage

dMT(X, Lambda, SigmaR, SigmaC, nu, log = FALSE)
rMT(n, Lambda, SigmaR, SigmaC, nu)

Arguments

- \( X \): Argument to the density function. Either a \( p \times q \) matrix or a \( p \times q \times n \) array.
- \( \text{Lambda} \): Mean parameter. Either a \( p \times q \) matrix or a \( p \times q \times n \) array.
- \( \text{SigmaR} \): Between-row covariance matrix. Either a \( p \times p \) matrix or a \( p \times p \times n \) array.
- \( \text{SigmaC} \): Between-column covariance matrix. Either a \( q \times q \) matrix or a \( q \times q \times n \) array.
- \( \nu \): Degrees-of-freedom parameter. A scalar or vector of length \( n \).
- \( \text{log} \): Logical; whether or not to compute the log-density.
- \( n \): Integer number of random samples to generate.
Details

The Matrix-T distribution \( X \sim \text{Matrix-T}(\Lambda, \Sigma, \Psi, \nu) \) on a random matrix \( X_{p \times q} \) is the marginal distribution of \( X \) in \( (X, V) \sim \text{MNIW}(\Lambda, \Sigma, \Psi, \nu) \), where the Matrix-Normal Inverse-Wishart (MNIW) distribution is defined in \texttt{mniw}.

Value

A vector length \( n \) for density evaluation, or an array of size \( p \times q \times n \) for random sampling.

| MNIW | Generate samples from the Matrix-Normal Inverse-Wishart distribution. |

Description

Generate samples from the Matrix-Normal Inverse-Wishart distribution.

Usage

\[
\text{rMNIW}(n, \text{Lambda}, \text{Sigma}, \text{Psi}, \text{nu}, \text{prec} = \text{FALSE}) \\
\text{rmniw}(n, \text{Lambda}, \text{Omega}, \text{Psi}, \text{nu})
\]

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>number of samples.</td>
</tr>
<tr>
<td>( \text{Lambda} )</td>
<td>A mean matrix of size ( p \times q ) or an array of size ( p \times q \times n ). Defaults to matrix of zeros when missing.</td>
</tr>
<tr>
<td>( \text{Sigma} )</td>
<td>A row-wise variance or precision matrix of size ( p \times p ), or an array of size ( p \times p \times n ). Defaults to the identity matrix when missing.</td>
</tr>
<tr>
<td>( \text{Psi} )</td>
<td>A scale matrix of size ( q \times q ), or an array of size ( q \times q \times n ). Defaults to identity matrix when missing.</td>
</tr>
<tr>
<td>( \text{nu} )</td>
<td>Scalar degrees-of-freedom parameter.</td>
</tr>
<tr>
<td>( \text{prec} )</td>
<td>Logical; whether or not ( \text{Sigma} ) is on the variance or precision scale.</td>
</tr>
<tr>
<td>( \text{Omega} )</td>
<td>A between-row precision matrix of size ( p \times p ), or an array of size ( p \times p \times n ). Defaults to the identity matrix when missing.</td>
</tr>
</tbody>
</table>

Details

The Matrix-Normal Inverse-Wishart (MNIW) distribution \( (X, V) \sim \text{MNIW}(\Lambda, \Sigma, \Psi, \nu) \) on random matrices \( X_{p \times q} \) and symmetric positive-definite \( V_{q \times q} \) is defined as

\[
V \sim \text{Inverse-Wishart}(\Psi, \nu) \\
X \mid V \sim \text{Matrix-Normal}(\Lambda, \Sigma, V),
\]
where the Matrix-Normal distribution is defined as the multivariate normal

\[ \text{vec}(X) \sim \mathcal{N}(\text{vec}(\Lambda), V \otimes \Sigma), \]

where \( \text{vec}(X) \) is a vector stacking the columns of \( X \), and \( V \otimes \Sigma \) denotes the Kronecker product.

\texttt{rmniw} is a convenience wrapper to \texttt{rMNIW(Sigma = Omega, prec = TRUE)}, for the common situation in Bayesian inference with conjugate priors when between-row variances are naturally parametrized on the precision scale.

**Value**

A list with elements:

- \textbf{X} Array of size \( p \times q \times n \) random samples from the Matrix-Normal component (see Details).
- \textbf{V} Array of size \( q \times q \times n \) of random samples from the Inverse-Wishart component.

**Examples**

```r
# problem dimensions
p <- 2
c <- 3
n <- 10 # number of samples

# parameter specification
Lambda <- matrix(rnorm(p*c),p,c) # single argument
Sigma <- rwish(n, Psi = diag(p), nu = p + rexp(1)) # vectorized argument
Psi <- rwish(n = 1, Psi = diag(c), nu = q + rexp(1)) # single argument
nu <- q + rexp(1)

# simulate n draws
rMNIW(n, Lambda = Lambda, Sigma = Sigma, Psi = Psi, nu = nu)
```

---

**MultiNormal**

The Multivariate Normal distribution.

**Description**

Density and random sampling for the Multivariate Normal distribution.

**Usage**

```r
dmNorm(x, mu, Sigma, log = FALSE)
rmNorm(n, mu, Sigma)
```
Arguments

x  Argument to the density function. A vector of length q or an n x q matrix.
mu Mean vector(s). Either a vector of length q or an n x q matrix. If missing defaults to a vector of zeros.
Sigma Covariance matrix or matrices. Either a q x q matrix or a q x q x n array. If missing defaults to the identity matrix.
log Logical; whether or not to compute the log-density.
n Integer number of random samples to generate.

Value

A vector for densities, or a n x q matrix for random sampling.

Examples

# Parameter specification
q <- 4 # number of dimensions
mu <- 1:q # mean vector
V <- toeplitz(exp(-seq(1:q))) # variance matrix

# Random sample
n <- 100
X <- rmNorm(n, mu, V)

dmNorm(X, mu, V, log = TRUE)

Description

Sample from the conditional parameter distribution given the data and hyperparameters of the Multivariate-Normal Random-Effects (mNormRE) model (see Details).

Usage

rRxNorm(n, x, V, lambda, Sigma)

Arguments

n Integer number of random samples to generate.
x Data observations. Either a vector of length q or a n x q matrix. In the latter case each row is a different vector.
V Observation variances. Either a matrix of size q x q or a q x q x n array.
lambda  Prior means. Either a vector of length q or an n x q matrix. In the latter case each row is a different mean. Defaults to zeros.

Sigma  Prior variances. Either a matrix of size q x q or a q x q x n array. Defaults to identity matrix.

Details

Consider the hierarchical multivariate normal model
\[
\mu \sim N(\lambda, \Sigma) \\
x \mid \mu \sim N(\mu, V).
\]

The Multivariate-Normal Random-Effects model \( \mu \sim RxNorm(x, V, \lambda, \Sigma) \) on the random vector \( \mu_q \) is defined as the posterior distribution \( p(\mu \mid x, \lambda, \Sigma) \). This distribution is multivariate normal; for the mathematical specification of its parameters please see vignette("mniw-distributions", package = "mniw").

Examples

```r
# data specification
q <- 5
y <- rnorm(q)
V <- rwish(1, diag(q), q+1)

# prior specification
lambda <- rep(0, q)
A <- diag(q)
n <- 10

# random sampling
rRxNormLM(n, y, V, lambda, A)
```

RxNormLM  Bayesian inference for a random-effects regression model.

Description

Gibbs sampler for posterior distribution of parameters and hyperparameters of a multivariate normal random-effects linear regression model called RxNormLM (see Details).

Usage

```r
RxNormLM(
  nsamples,
  Y,
  V,
  X,
  prior = NULL,
  init,
  burn,
  updateHyp = TRUE,
)```
storeHyp = TRUE,
updateRX = TRUE,
storeRX = FALSE
)

Arguments

nsamples: number of posterior samples to draw.

Y: N x q matrix of responses.

V: Either a q x q variance matrix or an q x q x N array of such matrices.

X: N x p matrix of covariates.

prior: parameters of the prior MNIW distribution on the hyperparameters (see Details).

init: (optional) list with elements Beta, Sigma, and Mu providing the initial values for these. Default values are Beta = matrix(0, p, q), Sigma = diag(q), and Mu = Y.

burn: integer number of burn-in samples, or fraction of nsamples to prepend as burn-in.

updateHyp, storeHyp: logical. Whether or not to update/store the hyperparameter draws.

updateRX, storeRX: logical. Whether or not to update/store the random-effects draws.

Details

The RxNormLM model is given by

\[ y_i \mid \mu_i \sim_i i\text{d}N(\mu_i, V_i) \]

\[ \mu_i \mid \beta, \Sigma \sim i\text{nd}N(x_i' \beta, \Sigma) \]

\[ \beta, \Sigma \sim MNIW(\Lambda, \Omega^{-1}, \Psi, \nu), \]

where \( y_i \) and \( \mu_i \) are response and random-effects vectors of length \( q \), \( x_i \) are covariate vectors of length \( p \), and \( (\beta, \Sigma) \) are hyperparameter matrices of size \( p \times q \) and \( q \times q \).

The MNIW prior distribution is given by a list with elements Lambda, Omega, Psi, and nu. If any of these is NULL or missing, the default value is 0. Note that Omega == 0 gives a Lebesgue prior to \( \beta \).

Value

A list with (potential) elements:

Beta: An p x q x nsamples array of regression coefficient iterations (if storeHyp == TRUE)

Sigma: An q x q x nsamples array of regression variance matrices (if storeHyp == TRUE)

Mu: An n x q x nsamples array of random effects (if storeRX == TRUE)
Examples

# problem dimensions
n <- sample(10:20,1) # number of observations
p <- sample(1:4,1) # number of covariates
q <- sample(1:4,1) # number of responses

# hyperparameters
Lambda <- rMNorm(1, Lambda = matrix(0, p, q))
Omega <- crossprod(rMNorm(1, Lambda = matrix(0, p, p)))
Psi <- crossprod(rMNorm(1, Lambda = matrix(0, q, q)))
nu <- rexp(1) + (q+1)
prior <- list(Lambda = Lambda, Omega = Omega, Psi = Psi, nu = nu)

# random-effects parameters
BSig <- rmniw(1, Lambda = Lambda, Omega = Omega, Psi = Psi, nu = nu)
Beta <- BSig$X
Sigma <- BSig$V

# design matrix
X <- rMNorm(1, matrix(0, n, p))

# random-effects themselves
Mu <- rmNorm(n, X %*% Beta, Sigma)

# generate response data
V <- rwish(n, Psi = diag(q), nu = q+1) # error variances
Y <- rmNorm(n, mu = Mu, Sigma = V) # responses

# visual checks for each component of Gibbs sampler
# sample from p(Mu | Beta, Sigma, Y)
nsamples <- 1e5
out <- RxNormLM(nsamples,
                Y = Y, V = V, X = X,
                prior = prior,
                init = list(Beta = Beta, Sigma = Sigma, Mu = Mu),
                burn = floor(nsamples/10),
                updateHyp = FALSE,
                storeHyp = FALSE,
                updateRX = TRUE,
                storeRX = TRUE)

# conditional distribution is RxNorm:
iObs <- sample(n, 1) # pick an observation at random
# calculate the RxNorm parameters
G <- Sigma %*% solve(V[,iObs] + Sigma)
xB <- c(X[iObs,,drop=FALSE] %*% Beta)
muRx <- G %*% (Y[iObs,] - xB) + xB
SigmaRx <- G %*% V[,iObs]

# a' * mu_i is univariate normal with known mean and variance:
a <- rnorm(q) # arbitrary vector
amui <- crossprod(a, out$Mu[iObs,])  # a' * mu_i

hist(amui, breaks = 100, freq = FALSE,
     xlab = "", main = expression("Histogram of "*a^T*mu[i]))
curve(dnorm(x, mean = sum(a * muRx),
       sd = sqrt(crossprod(a, SigmaRx %*% a)[1]),
       add = TRUE, col = "red")
legend("topright",
       legend = c("Observed", "Expected"),
       lwd = c(NA, 2), pch = c(22, NA), seg.len = 1.5,
       col = c("black", "red"), bg = c("white", NA))

# sample from p(Beta, Sigma | Mu, Y)
nsamples <- 1e5
out <- RxNormLM(nsamples,
     Y = Y, V = V, X = X,
     prior = prior,
     init = list(Beta = Beta, Sigma = Sigma, Mu = Mu),
     burn = floor(nsamples/10),
     updateHyp = TRUE,
     storeHyp = TRUE,
     updateRX = FALSE,
     storeRX = FALSE)

# conditional distribution is MNIW:
# calculate the MNIW parameters
OmegaHat <- crossprod(X) + Omega
LambdaHat <- solve(OmegaHat, crossprod(X, Mu) + Omega %*% Lambda)
PsiHat <- Psi + crossprod(Mu) + crossprod(Lambda, Omega %*% Lambda)
PsiHat <- PsiHat - crossprod(LambdaHat, OmegaHat %*% LambdaHat)
nuHat <- nu + n

# a' Sigma^{-1} a is chi^2 with known parameters:
a <- rnorm(q)
aSiga <- drop(crossprodV(a, V = out$Sigma, inverse = TRUE))
sigX <- crossprod(a, solve(PsiHat, a))[1]
hist(aSiga, breaks = 100, freq = FALSE,
     xlab = "", main = expression("Histogram of "*a^T*Sigma^{-1}*a))
curve(dchisq(x/sigX, df = nuHat)/sigX, add = TRUE, col = "red")
legend("topright",
       legend = c("Observed", "Expected"),
       lwd = c(NA, 2), pch = c(22, NA), seg.len = 1.5,
       col = c("black", "red"), bg = c("white", NA))

# a' Beta b is student-t with known parameters:
a <- rnorm(p)
b <- rnorm(q)
# vectorized calculations
aBetab <- crossprodV(X = aperm(out$Beta, c(2,1,3)),
                     Y = b, V = diag(q))  # Beta b
$$aBeta_b <- \text{drop(\text{crossprod}(X = a, Y = aBeta_b, V = \text{diag}(p)))}$$  # a' Beta b
# student-t parameters
muT <- \text{crossprod}(a, LambdaHat \times b)[1]
nuT <- \text{nuHat} - q + 1
sigmaT <- \text{crossprod}(a, V = OmegaHat, inverse = \text{TRUE})[1]
sigmaT <- sigmaT * \text{crossprod}(b, V = PsiHat)[1]
sigmaT <- \sqrt{\text{sigmaT} / nuT}

\text{hist(aBeta_b, breaks = 100, freq = FALSE,}
  xlab = '', main = expression("Histogram of a^T Beta a"))
\text{curve(dt((x-muT)/sigmaT, df = nuT)/sigmaT, add = TRUE, col = "red")}
\text{legend("topright",}
  legend = c("Observed", "Expected"),
  lwd = c(NA, 2), pch = c(22, NA), seg.len = 1.5,
  col = c("black", "red"), bg = c("white", NA))

---

**Wishart**

Wishart and Inverse-Wishart distributions.

**Description**

Densities and random sampling for the Wishart and Inverse-Wishart distributions.

**Usage**

- \text{dwish}(X, Psi, nu, log = \text{FALSE})
- \text{rwish}(n, Psi, nu)
- \text{diwish}(X, Psi, nu, log = \text{FALSE})
- \text{riwish}(n, Psi, nu)
- \text{dwishart}(X, Psi, nu, inverse = \text{FALSE}, log = \text{FALSE})
- \text{rwishart}(n, Psi, nu, inverse = \text{FALSE})

**Arguments**

- \text{X}  
  Argument to the density function. Either a q \times q matrix or a q \times q \times n array.
- \text{Psi}  
  Scale parameter. Either a q \times q matrix or a q \times q \times n array.
- \text{nu}  
  Degrees-of-freedom parameter. A scalar or vector of length n.
- \text{log}  
  Logical; whether or not to compute the log-density.
- \text{n}  
  Integer number of random samples to generate.
- \text{inverse}  
  Logical; whether or not to use the Inverse-Wishart distribution.
Details

The Wishart distribution $X \sim \text{Wishart}(\Psi, \nu)$ on a symmetric positive-definite random matrix $X$ of size $q \times q$ has PDF

$$f(X | \Psi, \nu) = \frac{|X|^{(\nu-q-1)/2} \exp\left\{ -\text{tr}(\Psi^{-1}X)/2 \right\}}{2^{\nu q/2} |\Psi|^{\nu/2} \Gamma_q(\nu/2)}$$

where $\Gamma_q(\alpha)$ is the multivariate gamma function,

$$\Gamma_q(\alpha) = \pi^{q(q-1)/4} \prod_{i=1}^{q} \Gamma(\alpha + (1 - i)/2).$$

The Inverse-Wishart distribution $X \sim \text{Inverse-Wishart}(\Psi, \nu)$ is defined as $X^{-1} \sim \text{Wishart}(\Psi^{-1}, \nu)$.

dwish and diwish are convenience wrappers for dwishart, and similarly rwish and riwish are wrappers for rwishart.

Value

A vector length $n$ for density evaluation, or an array of size $q \times q \times n$ for random sampling.

Examples

```r
# Random sampling
n <- 1e5
q <- 3
Psi1 <- crossprod(matrix(rnorm(q^2), q, q))
nu <- q + runif(1, 0, 5)
X1 <- rwish(n, Psi1, nu) # Wishart

# plot it
plot_fun <- function(X) {
  q <- dim(X)[1]
  par(mfrow = c(q, q))
  for(ii in 1:q) {
    for(jj in 1:q) {
      hist(X[ii,jj,], breaks = 100, freq = FALSE, 
           xlab = "", main = parse(text = paste0("X[", ii, jj, "]")))
    }
  }
}
plot_fun(X1)

# "vectorized" scale parameter
Psi2 <- 5 * Psi1
vPsi <- array(c(Psi1, Psi2), dim = c(q, q, n))
X2 <- rwish(n, Psi = vPsi, nu = nu)
plot_fun(X2)
```
# Inverse-Wishart
X3 <- riwish(n, Psi2, nu)
plot_fun(X3)

# log-density calculation for sampled values
par(mfrow = c(1,1))
hist(dwish(X2, vPsi, nu, log = TRUE),
     breaks = 100, freq = FALSE, xlab = "",
     main = expression("log-p"*(X[2]" | "*list(Psi,nu)))))
Index

* datasets
  Hospitals, 3

crossprodV, 3

diwish (Wishart), 13
dMN (MatrixNormal), 4
dmNorm (MultiNormal), 7
dMT (MatrixT), 5
dwish (Wishart), 13
dwishart (Wishart), 13

Hospitals, 3

MatrixNormal, 4
MatrixT, 5
MNIW, 6

mniw, 6
mniw-package, 2

MultiNormal, 7

riwish (Wishart), 13
rMNIW (MNIW), 6
rmniw (MNIW), 6
rMN (MatrixNormal), 4
rmNorm (MultiNormal), 7

rMT (MatrixT), 5
rR (MatrixT), 8
rwish (Wishart), 13
rwishart (Wishart), 13
R (MatrixT), 8

Wishart, 13