Package ‘MultiRNG’

March 29, 2020

Type Package
Title Multivariate Pseudo-Random Number Generation
Version 1.2.3
Date 2020-03-29
Author Hakan Demirtas, Rawan Allozi, Ran Gao
Maintainer Ran Gao <rgao8@uic.edu>
License GPL-2 | GPL-3
NeedsCompilation no
Repository CRAN
Date/Publication 2020-03-29 21:40:02 UTC

R topics documented:

MultiRNG-package ......................................................... 2
draw.correlated.binary ................................................. 3
draw.d.variate.normal .................................................... 4
draw.d.variate.t .......................................................... 5
draw.d.variate.uniform .................................................. 6
draw.dirichlet ............................................................. 7
draw.dirichlet.multinomial ............................................ 8
draw.inv.wishart ......................................................... 9
draw.multinomial ......................................................... 10
draw.multivariate.hypergeometric ................................ 11
draw.multivariate.laplace ............................................ 12
draw.wishart ............................................................. 13
generate.point.in.sphere ............................................. 14
loc.min ................................................................. 14

Index 16
Description

This package implements the algorithms described in Demirtas (2004) for pseudo-random number generation of 11 multivariate distributions. The following multivariate distributions are available: Normal, t, Uniform, Bernoulli, Hypergeometric, Beta (Dirichlet), Multinomial, Dirichlet-Multinomial, Laplace, Wishart, and Inverted Wishart.

This package contains 11 main functions and 2 auxiliary functions. The methodology for each random-number generation procedure varies and each distribution has its own function. For multivariate normal, `draw.d.variate.normal` employs the Cholesky decomposition and a vector of univariate normal draws and for multivariate t, `draw.d.variate.t` employs the Cholesky decomposition and a vector of univariate normal and chi-squared draws. `draw.d.variate.uniform` is based on cdf of multivariate normal deviates (Falk, 1999) and `draw.correlated.binary` generates correlated binary variables using an algorithm developed by Park, Park and Shin (1996) and makes use of the auxiliary function `loc.min`. `draw.multivariate.hypergeometric` employs sequential generation of succeeding conditionals which are univariate hypergeometric. Furthermore, `draw.dirichlet` uses the ratios of gamma variates with a common scale parameter and `draw.multinomial` generates data via sequential generation of marginals which are binomials. `draw.dirichlet.multinomial` is a mixture distribution of a multinomial that is a realization of a random variable having a Dirichlet distribution. `draw.multivariate.laplace` is based on generation of a point s on the d-dimensional sphere and utilizes the auxiliary function `generate.point.in.sphere`. `draw.wishart` and `draw.inv.wishart` employs Wishart variates that follow d-variate normal distribution.

Details

| Package:   | MultiRNG   |
| Type:      | Package    |
| Version:   | 1.2.3      |
| Date:      | 2020-03-29 |
| License:   | GPL-2 | GPL-3 |

Author(s)

Hakan Demirtas, Rawan Allozi, Ran Gao
Maintainer: Ran Gao <rgao8@uic.edu>

References


---

draw.correlated.binary

*Generation of Correlated Binary Data*

**Description**

This function implements pseudo-random number generation for a multivariate Bernoulli distribution (correlated binary data).

**Usage**

draw.correlated.binary(no.row, d, prop.vec, corr.mat)

**Arguments**

- `no.row` Number of rows to generate.
- `d` Number of variables to generate.
- `prop.vec` Vector of means.
- `corr.mat` Correlation matrix.

**Value**

A `no.row × d` matrix of generated data.

**References**


**See Also**

loc.min

**Examples**

```r
cmat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
propvec=c(0.3,0.5,0.7)

mydata=draw.correlated.binary(no.row=1e5,d=3,prop.vec=propvec,corr.mat=cmat)
apply(mydata,2,mean)-propvec
cor(mydata)-cmat
```
draw.d.variate.normal  Pseudo-Random Number Generation under Multivariate Normal Distribution

Description

This function implements pseudo-random number generation for a multivariate normal distribution with pdf

\[ f(x|\mu, \Sigma) = c \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right) \]

for \(-\infty < x < \infty\) and \(c = (2\pi)^{-d/2} |\Sigma|^{-1/2}\), \(\Sigma\) is symmetric and positive definite, where \(\mu\) and \(\Sigma\) are the mean vector and the variance-covariance matrix, respectively.

Usage

\[
\text{draw.d.variate.normal}(\text{no.row}, d, \text{mean.vec}, \text{cov.mat})
\]

Arguments

- `no.row`: Number of rows to generate.
- `d`: Number of variables to generate.
- `mean.vec`: Vector of means.
- `cov.mat`: Variance-covariance matrix.

Value

A `no.row` x `d` matrix of generated data.

Examples

\[
\begin{align*}
\text{cmat} &\leftarrow \text{matrix}(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), \text{nrow}=3, \text{ncol}=3) \\
\text{meanvec} &\leftarrow c(0,3,7) \\
\text{mydata} &\leftarrow \text{draw.d.variate.normal}(\text{no.row}=1e5, d=3, \text{mean.vec}=\text{meanvec}, \text{cov.mat}=\text{cmat}) \\
\text{apply} &\leftarrow (\text{mydata}, 2, \text{mean}) \leftarrow \text{meanvec} \\
\text{cor} &\leftarrow (\text{mydata}) \leftarrow \text{cmat}
\end{align*}
\]
draw.d.variate.t

Pseudo-Random Number Generation under Multivariate $t$ Distribution

Description

This function implements pseudo-random number generation for a multivariate $t$ distribution with pdf

$$f(x|\mu, \Sigma, \nu) = c \left( 1 + \frac{1}{\nu} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)^{-(\nu+d)/2}$$

for $-\infty < x < \infty$ and $c = \frac{\Gamma((\nu+d)/2)}{\Gamma(\nu/2)\Gamma(d/2)|\Sigma|^{-1/2}}$, $\Sigma$ is symmetric and positive definite, $\nu > 0$, where $\mu$, $\Sigma$, and $\nu$ are the mean vector, the variance-covariance matrix, and the degrees of freedom, respectively.

Usage

draw.d.variate.t(dof, no.row, d, mean.vec, cov.mat)

Arguments

dof Degrees of freedom.
no.row Number of rows to generate.
d Number of variables to generate.
mean.vec Vector of means.
cov.mat Variance-covariance matrix.

Value

A $\text{no.row} \times d$ matrix of generated data.

Examples

cmat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
meanvec=c(0,3,7)
mydata=draw.d.variate.t(dof=5, no.row=1e5, d=3, mean.vec=meanvec, cov.mat=cmat)
apply(mydata,2,mean)-meanvec
cor(mydata)-meanvec
draw.d.variate.uniform

Pseudo-Random Number Generation under Multivariate Uniform Distribution

Description

This function implements pseudo-random number generation for a multivariate uniform distribution with specified mean vector and covariance matrix.

Usage

draw.d.variate.uniform(no.row,d,cov.mat)

Arguments

no.row Number of rows to generate.
d Number of variables to generate.
cov.mat Variance-covariance matrix.

Value

A no.row × d matrix of generated data.

References


Examples

cmat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
mydata=draw.d.variate.uniform(no.row=1e5,d=3,cov.mat=cmat)
apply(mydata,2,mean)-rep(0.5,3)
cor(mydata)-cmat
**draw.dirichlet**

**Pseudo-Random Number Generation under Multivariate Beta (Dirichlet) Distribution**

**Description**

This function implements pseudo-random number generation for a multivariate beta (Dirichlet) distribution with pdf

\[
f(x|\alpha_1, \ldots, \alpha_d) = \frac{\Gamma\left(\sum_{j=1}^{d} \alpha_j\right)}{\prod_{j=1}^{d} \Gamma(\alpha_j)} \prod_{j=1}^{d} x_j^{\alpha_j-1}
\]

for \(\alpha_j > 0, x_j \geq 0, \) and \(\sum_{j=1}^{d} x_j = 1\), where \(\alpha_1, \ldots, \alpha_d\) are the shape parameters and \(\beta\) is a common scale parameter.

**Usage**

`draw.dirichlet(no.row, d, alpha, beta)`

**Arguments**

- **no.row** Number of rows to generate.
- **d** Number of variables to generate.
- **alpha** Vector of shape parameters.
- **beta** Scale parameter common to \(d\) variables.

**Value**

A \(no.row \times d\) matrix of generated data.

**Examples**

```r
alpha.vec=c(1,3,4,4)
mydata=draw.dirichlet(no.row=1e5, d=4, alpha=alpha.vec, beta=2)
apply(mydata, 2, mean) - alpha.vec/sum(alpha.vec)
```
**draw.dirichlet.multinomial**

*Pseudo-Random Number Generation under Dirichlet-Multinomial Distribution*

**Description**

This function implements pseudo-random number generation for a Dirichlet-multinomial distribution. This is a mixture distribution that is multinomial with parameter $\theta$ that is a realization of a random variable having a Dirichlet distribution with shape vector $\alpha$. $N$ is the sample size and $\beta$ is a common scale parameter.

**Usage**

```r
draw.dirichlet.multinomial(no.row, d, alpha, beta, N)
```

**Arguments**

- `no.row`: Number of rows to generate.
- `d`: Number of variables to generate.
- `alpha`: Vector of shape parameters.
- `beta`: Scale parameter common to $d$ variables.
- `N`: Sample size.

**Value**

A $no.row \times d$ matrix of generated data.

**See Also**

`draw.dirichlet, draw.multinomial`

**Examples**

```r
alpha.vec=c(1,3,4,4); N=3
mydata=draw.dirichlet.multinomial(no.row=1e5, d=4, alpha=alpha.vec, beta=2, N=3)
apply(mydata, 2, mean) - N*alpha.vec/sum(alpha.vec)
```
draw.inv.wishart

Pseudo-Random Number Generation under Inverted Wishart Distribution

Description

This function implements pseudo-random number generation for an inverted Wishart distribution with pdf

\[
f(x|\nu, \Sigma) = (2^{d/2} \pi^{d(d-1)/4} \prod_{i=1}^{d} \Gamma((\nu + 1 - i)/2))^{-1} |\Sigma|^{\nu/2} |x|^{-(\nu+d+1)/2} \exp\left(-\frac{1}{2} tr(\Sigma x^{-1})\right)
\]

\(x\) is positive definite, \(\nu \geq d\), and \(\Sigma^{-1}\) is symmetric and positive definite, where \(\nu\) and \(\Sigma^{-1}\) are the degrees of freedom and the inverse scale matrix, respectively.

Usage

\[
draw.inv.wishart(no.row,d,nu,inv.sigma)
\]

Arguments

- **no.row**: Number of rows to generate.
- **d**: Number of variables to generate.
- **nu**: Degrees of freedom.
- **inv.sigma**: Inverse scale matrix.

Value

A \(no.row \times d^2\) matrix containing Wishart deviates in the form of rows. To obtain the Inverted-Wishart matrix, convert each row to a matrix where rows are filled first.

See Also

- `draw.wishart`

Examples

```r
mymat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
draw.inv.wishart(no.row=1e5,d=3,nu=5,inv.sigma=mymat)
```
**draw.multinomial**

*Pseudo-Random Number Generation under Multivariate Multinomial Distribution*

**Description**

This function implements pseudo-random number generation for a multivariate multinomial distribution with pdf

\[ f(x | \theta_1, ..., \theta_d) = \frac{N!}{\prod x_j!} \prod_{j=1}^d \theta_j^{x_j} \]

for \(0 < \theta_j < 1, x_j \geq 0, \) and \( \sum_{j=1}^d x_j = N\), where \(\theta_1, ..., \theta_d\) are cell probabilities and \(N\) is the size.

**Usage**

```r
draw.multinomial(no.row, d, theta, N)
```

**Arguments**

- `no.row` Number of rows to generate.
- `d` Number of variables to generate.
- `theta` Vector of cell probabilities.
- `N` Sample Size. Must be at least 2.

**Value**

A `no.row x d` matrix of generated data.

**Examples**

```r
theta.vec=c(0.3, 0.3, 0.25, 0.15); N=4
mydata=draw.multinomial(no.row=1e5, d=4, theta=c(0.3, 0.3, 0.25, 0.15), N=4)
apply(mydata, 2, mean) - N*theta.vec
```
**draw.multivariate.hypergeometric**

*Pseudo-Random Number Generation under Multivariate Hypergeometric Distribution*

**Description**

This function implements pseudo-random number generation for a multivariate hypergeometric distribution.

**Usage**

```
draw.multivariate.hypergeometric(no.row,d,mean.vec,k)
```

**Arguments**

- `no.row` Number of rows to generate.
- `d` Number of variables to generate.
- `mean.vec` Number of items in each category.
- `k` Number of items to be sampled. Must be a positive integer.

**Value**

A `no.row` × `d` matrix of generated data.

**References**


**Examples**

```r
meanvec=c(10,10,12) ; myk=5
mydata=draw.multivariate.hypergeometric(no.row=1e5,d=3,mean.vec=meanvec,k=myk)
apply(mydata,2,mean)-myk*meanvec/sum(meanvec)
```
draw.multivariate.laplace

*Pseudo-Random Number Generation under Multivariate Laplace Distribution*

**Description**

This function implements pseudo-random number generation for a multivariate Laplace (double exponential) distribution with pdf

\[ f(x|\mu, \Sigma, \gamma) = c \exp(-((x - \mu)^T \Sigma^{-1} (x - \mu))^\gamma/2) \]

for \(-\infty < x < \infty\) and \(c = \frac{\gamma^{d/2}}{2\pi^{d/2} |\Sigma|^{-1/2}}\), \(\Sigma\) is symmetric and positive definite, where \(\mu\), \(\Sigma\), and \(\gamma\) are the mean vector, the variance-covariance matrix, and the shape parameter, respectively.

**Usage**

```r
draw.multivariate.laplace(no.row, d, gamma, mu, Sigma)
```

**Arguments**

- `no.row`: Number of rows to generate.
- `d`: Number of variables to generate.
- `gamma`: Shape parameter.
- `mu`: Vector of means.
- `Sigma`: Variance-covariance matrix.

**Value**

A `no.row x d` matrix of generated data.

**References**


**See Also**

- `generate.point.in.sphere`

**Examples**

```r
cmat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
mu.vec=c(0,3,7)
mydata=draw.multivariate.laplace(no.row=1e5, d=3, gamma=2, mu=mu.vec, Sigma=cmat)

apply(mydata,2,mean)-mu.vec
cor(mydata)-cmat
```
draw.wishart

*Pseudo-Random Number Generation under Wishart Distribution*

**Description**

This function implements pseudo-random number generation for a Wishart distribution with pdf

\[
  f(x|\nu, \Sigma) = \left(\frac{2^{\nu d/2}}{\pi^{d(d-1)/4}}\right) \prod_{i=1}^{d} \Gamma\left((\nu + 1 - i)/2\right) |\Sigma|^{-\nu/2} |x|^{(\nu - d - 1)/2} \exp\left(-\frac{1}{2} tr(\Sigma^{-1}x)\right)
\]

\(x\) is positive definite, \(\nu \geq d\), and \(\Sigma\) is symmetric and positive definite, where \(\nu\) and \(\Sigma\) are positive definite and the scale matrix, respectively.

**Usage**

\[
  \text{draw.wishart}(\text{no.row}, d, \nu, \sigma)
\]

**Arguments**

- **no.row**: Number of rows to generate.
- **d**: Number of variables to generate.
- **nu**: Degrees of freedom.
- **sigma**: Scale matrix.

**Value**

A \(\text{no.row} \times d^2\) matrix of Wishart deviates in the form of rows. To obtain the Wishart matrix, convert each row to a matrix where rows are filled first.

**See Also**

- **draw.d.variate.normal**

**Examples**

```r
mymat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
draw.wishart(no.row=1e5,d=3,nu=5,sigma=mymat)
```
generate.point.in.sphere

*Point Generation for a Sphere*

**Description**

This function generates s points on a d-dimensional sphere.

**Usage**

`generate.point.in.sphere(no.row,d)`

**Arguments**

- `no.row`: Number of rows to generate.
- `d`: Number of variables to generate.

**Value**

A `no.row × d` matrix of coordinates of points in sphere.

**References**


**Examples**

`generate.point.in.sphere(no.row=1e5,d=3)`

---

**loc.min**

*Minimum Location Finder*

**Description**

This function identifies the location of the minimum value in a square matrix.

**Usage**

`loc.min(my.mat,d)`

**Arguments**

- `my.mat`: A square matrix.
- `d`: Dimensions of the matrix.
Value

A vector containing the row and column number of the minimum value.

Examples

cmat<-matrix(c(1,0.2,0.3,0.2,1,0.2,0.3,0.2,1), nrow=3, ncol=3)
loc.min(my.mat=cmat, d=3)
Index

draw.correlated.binary, 3
draw.d.variate.normal, 4, 13
draw.d.variate.t, 5
draw.d.variate.uniform, 6
draw.dirichlet, 7, 8
draw.dirichlet.multinomial, 8
draw.inv.wishart, 9
draw.multinomial, 8, 10
draw.multivariate.hypergeometric, 11
draw.multivariate.laplace, 12
draw.wishart, 9, 13

generate.point.in.sphere, 12, 14

loc.min, 3, 14

MultiRNG (MultiRNG-package), 2
MultiRNG-package, 2