Package 'fastMPRG'

January 6, 2023

Type Package

Title Fast creation of multivariate data

Version 1.0

Date 2023-01-05

Author Ye Cong , Xu Zicheng

Maintainer Ye Cong<yecong@ruc.edu.cn> and
Xu Zicheng<1223237249@qq.com>

Description You can use this package to generate multivariate data.
And the data distribution type contains 'norm','unif','exp','beta' and 'LN'.
You can also use this package to generate two dimensions data which has different distribution.And the distribution type contains 'chi-square','unif','norm'.

License MIT + file LICENSE

Imports Rcpp (>= 1.0.9)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.3

Encoding UTF-8

Depends cubature, ggplot2

NeedsCompilation yes

R topics documented:

`fastMPRG-package` *Fast creation of multivariate data*

Description

You can use this package to generate multivariate data. And the data distribution type contains 'norm','unif','exp','beta' and 'LN'. You can also use this package to generate two dimensions data which has different distribution.And the distribution type contains 'chi-square','unif','norm'.

Details

The DESCRIPTION file:

Package:	fastMPRG
Type:	Package
Title:	Fast creation of multivariate data
Version:	1.0
Date:	2023-01-05
Author:	Cong Ye and Zicheng Xu
Maintainer:	Cong Ye<yecong@ruc.edu.cn> and Zicheng Xu<1223237249@qq.com>
Description:	You can use this package to generate multivariate data. And the data distribution type contains 'norm','u
License:	MIT + file LICENSE
Imports:	Rcpp (>= 1.0.9)
LinkingTo:	Rcpp, RcppArmadillo
RoxygenNote:	7.2.3
Encoding:	UTF-8
Depends:	cubature, ggplot2
Archs:	x64

Index of help topics:

	Given Correlation Structure.
nor_unif	The Data Generation of Normal Distribution and Uniform Distribution

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Cong Ye and Zicheng Xu

Maintainer: Cong Ye<yecong@ruc.edu.cn> and Zicheng Xu<1223237249@qq.com>

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ <pkg> ~~

Examples

```
~~ simple examples of the most important functions ~~
```

ARStructure *Generate a Correlation Matrix that has AR Structure.*

Description

This function generates a correlation matrix that has an autoregressive structure. That is, $R = (r_{ij})_{n\times n}$, wehere $r_{ij} = \rho^{|i-j|}$.

Usage

```
ARStructure(rho,d)
```

Arguments

rho	This is the basis autocorrelation coefficient we implement in the correlation matrix.
d	This is the dimensionality of our correlation matrix.

Value

A $d \times d$ AR structure correlation matrix.

Examples

```
A = ARStructure(0.9,5)
```

chi_nor *The Data Generation of Chi-square Distribution and Normal Distribution*

Description

The generation of two related sequences follows chi-square distribution and normal distribution respectively

Usage

```
chi_nor(n, corr, rd = 1, mean = 0, sigma = 1)
```

Arguments

n	number of generation data from each distribution
corr	a number, the correlation between the two distribution
rd	degrees of freedom of chi square distribution
mean	the mean of normal distribution
sigma	the standard deviation of normal distribution

Value

a matrix which the first column obey chi-square distribution and the second column obey normal distribution

Examples

```
my_mat = chi_nor(20000,0.6,rd = 5,mean = 3,sigma = 2)
new_mat = as.data.frame(my_mat)
cor(new_mat)
cov(new_mat)
```

chi_unif *The Data Generation of Chi-square Distribution and Uniform Distribution*

Description

The generation of two related sequences follows chi-square distribution and uniform distribution respectively

Usage

```
chi_unif(n, corr, rd = 1, min = 0, max = 1)
```

Arguments

n	number of generation data from each distribution
corr	a number, the correlation between the two distribution
rd	degrees of freedom of chi square distribution
min	the minimum number of uniform distribution
max	the maximum number of uniform distribution

Value

a matrix which the first column obey chi-square distribution and the second column obey uniform distribution

Examples

```
my_mat = chi_unif(20000,0.4,rd = 5, min = 2,max = 5)
new_mat = as.data.frame(my_mat)
cor(new_mat)
cov(new_mat)
```

---

corr *Calculate the Pearson correlation coefficient of each variables.*

Description

'corr' returns the Pearson correlation coefficient matrix to detect whether our simulated data meet the required correlation structure.

Usage

```
corr(data)
```

Arguments

data	The data generated by our methods (return values).

Value

A Pearson Correlation Coefficient Matrix.

Examples

```
A = ARStructure(0.9,5)
a = fastunif(A,50000)
corr(a)
```

`draw_hist` *Draw the histogram of each columns of our data.*

Description

'draw_hist' returns a ggplot2 figure with histogram and theoretical density curve to detect whether our generated data meet the required marginal distribution shape.

Usage

```
draw_hist(df,col,bin,type,xlim = c(0,1))
```

Arguments

`df`	The data generated by our methods (return values).
`col`	The columns number of certain data you want to plot. Starts from 1.
`bin`	The number of bins you want to display in your histogram.
`type`	The theoretical distribution name. This parameter only can be processed within 5 values right now: 'norm','unif','exp','beta', 'chi-square' and 'LN'.
`xlim`	To limit the display range of horizontal axis. Default is (0,1), which means only display the generated data from 0 to 1.

Value

A histogram with theoretical density curve.

Examples

```
A = ARStructure(0.9,5)
a = fastunif(A,50000)
draw_hist(a,1,100,'unif')
```

`fastbeta` *Generate Multivariate Beta Distribution Data Follow the Given Correlation Structure.*

Description

This function implements a fast pseudo random number generator for a multivariate beta distribution where every marginal distribution is $Beta(\frac{1}{n}, 1)$

Usage

```
fastbeta(rela_mat, n, row)
```

Arguments

rela_mat	The desired correlation matrix structure you want the generated data follow. This matrix must be positive definite.
n	is a parameter of beta distribution. See detail.
row	The observed sample size you want to generate.

Value

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

Examples

```
A = ARStructure(0.9,5)
n = 1/5
a = fastbeta(A,n,50000)
```

---

fastexp *Generate Multivariate Exponential Data Follow the Given Correlation Structure.*

Description

This function implements a fast pseudo random number generator for a multivariate exponential distribution where every marginal distribution has equal parameter λ.

Usage

```
fastexp(rela_mat, lambda, row)
```

Arguments

rela_mat	The desired correlation matrix structure you want the generated data follow. This matrix must be positive definite.
lambda	The parameter for the exponential distributions.
row	The observed sample size you want to generate.

Details

If $X \sim Exp(\lambda)$, then its pdf is :

$$f(x) = \lambda exp{-}\lambda x, x > 0$$

.

Value

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

Examples

```
A = ARStructure(0.9,5)
lambda = 2
a = fastexp(A,lambda,50000)
```

`fastLN` *Generate Multivariate Logarithmic Normal Data Follow the Given Correlation Structure.*

Description

This function implements a fast pseudo random number generator for a multivariate logarithmic normal distribution where every marginal distribution has equal variance.

Usage

```
fastLN(rela_mat, row, mu, sigma)
```

Arguments

`rela_mat`	The desired correlation matrix structure you want the generated data follow. This matrix must be positive definite.
`row`	The observed sample size you want to generate.
`mu`	The mean vector you want the generated data follow. It must have the same dimensionality of the correlation matrix.
`sigma`	The common marginal variance level. That is, we assume that the marginal distribution has identical variance.

Details

For $X_i \sim N(\mu, \sigma^2)$, we say $Y_i = e^{X_i}$ follows a logarithmic normal distribution.

Value

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

Examples

```
A = ARStructure(0.9,5)
mu = rep(0,5)
sigma = 1
a = fastLN(A,50000,mu,sigma)
```

`fastnorm` *Generate Multivariate Normal Distribuion Data Follow the Given Correlation Structure.*

Description

This function implements a fast pseudo random number generator for a multivariate normal distribution where every marginal

Usage

```
fastnorm(rela_mat, row, mu, sigma)
```

Arguments

rela_mat	the desired correlation matrix structure you want the generated data follow. This matrix must be positive definite.
row	the observed sample size you want to generate.
mu	a vector contain the respective normal distribution mean
sigma	a vector contain the respective normal distribution standard deviation

Value

a matrix which every column is a data from one kind normal distribution

Examples

```
A = ARStructure(0.9,5)
fastnorm(rela_mat = A,row = 100,mu = rep(0,5),sigma = rep(1,5))
```

---

fastunif — *Generate Multivariate Uniform Data Follow the Given Correlation Structure.*

Description

This function implements a fast pseudo random number generator for a multivariate uniform distribution where every marginal distribution is $Unif(0, 1)$.

Usage

```
fastunif(rela_mat, row)
```

Arguments

rela_mat	The desired correlation matrix structure you want the generated data follow. This matrix must be positive definite.
row	The observed sample size you want to generate.

Value

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

Examples

```
A = ARStructure(0.9,5)
a = fastunif(A,50000)
```

nor_unif *The Data Generation of Normal Distribution and Uniform Distribution*

Description

The generation of two related sequences follows normal distribution and uniform distribution respectively

Usage

```
nor_unif(n, corr, mean = 0, sigma = 1, min = 0, max = 1)
```

Arguments

n	number of generation data from each distribution
corr	a number, the correlation between the two distribution
mean	the mean of normal distribution
sigma	the standard deviation of normal distribution
min	the minimum number of uniform distribution
max	the maximum number of uniform distribution

Value

a matrix which the first column obey normal distribution, and the second column obey uniform distribution

Examples

```
my_mat = nor_unif(20000,0.9,mean = 3,sigma = 2,min = 2,max = 3)
new_mat = as.data.frame(my_mat)
cor(new_mat)
cov(new_mat)
```

Index