

Package ‘randcorr’

July 23, 2025

Type Package

Title Generate a Random $p \times p$ Correlation Matrix

Version 1.0

Date 2018-11-07

Author Daniel F. Schmidt [aut, cph, cre],
Enes Makalic [aut, cph]

Maintainer Daniel F. Schmidt <daniel.schmidt@monash.edu>

Description Implements the algorithm by Pourahmadi and Wang (2015) <[doi:10.1016/j.spl.2015.06.015](https://doi.org/10.1016/j.spl.2015.06.015)> for generating a random $p \times p$ correlation matrix. Briefly, the idea is to represent the correlation matrix using Cholesky factorization and $p(p-1)/2$ hyperspherical coordinates (i.e., angles), sample the angles from a particular distribution and then convert to the standard correlation matrix form. The angles are sampled from a distribution with pdf proportional to $\sin^k(\theta)$ ($0 < \theta < \pi$, $k \geq 1$) using the efficient sampling algorithm described in Enes Makalic and Daniel F. Schmidt (2018) <[doi:10.48550/arXiv.1809.05212](https://doi.org/10.48550/arXiv.1809.05212)>.

License GPL (≥ 3)

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-16 15:30:03 UTC

Contents

randcorr-package	2
randcorr	3
randcorr.sample.sink	4
Index	6

randcorr-package

The randcorr package

Description

This package contains a function to generate a random $p \times p$ correlation matrix. This function implements the algorithm by Pourahmadi and Wang [1] for generating a random $p \times p$ correlation matrix. Briefly, the idea is to represent the correlation matrix using Cholesky factorization and $p(p-1)/2$ hyperspherical coordinates (i.e., angles), sample the angles from a particular distribution and then convert to the standard correlation matrix form. The angles are sampled from a distribution with a probability density function proportional to $\sin^k(\theta)$ ($0 < \theta < \pi$, $k \geq 1$) using the efficient sampling algorithm described in [2].

Details

For usage, see the examples in `randcorr` and `randcorr.sample.sink`.

Note

To cite this package please reference:

Makalic, E. & Schmidt, D. F. An efficient algorithm for sampling from $\sin^k(x)$ for generating random correlation matrices arXiv:1809.05212, 2018 <https://arxiv.org/abs/1809.05212>

A MATLAB-compatible implementation of the sampler in this package can be obtained from:

<https://au.mathworks.com/matlabcentral/fileexchange/68810-randcorr>

Author(s)

Daniel Schmidt <daniel.schmidt@monash.edu>

Faculty of Information Technology, Monash University, Australia

Enes Makalic <emakalic@unimelb.edu.au>

Centre for Epidemiology and Biostatistics, The University of Melbourne, Australia

References

[1] Mohsen Pourahmadi and Xiao Wang, Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor, Statistics & Probability Letters, Volume 106, Pages 5-12, 2015.

[2] Enes Makalic and Daniel F. Schmidt An efficient algorithm for sampling from $\sin^k(x)$ for generating random correlation matrices, arXiv:1809.05212, 2018.

See Also

`randcorr`, `randcorr.sample.sink`

randcorr	<i>Generate a random $p \times p$ correlation matrix</i>
----------	---

Description

Generate a random $p \times p$ correlation matrix

Usage

`randcorr(p)`

Arguments

`p` A scalar positive integer denoting the size of the correlation matrix

Value

A random $p \times p$ correlation matrix

Details

This function implements the algorithm by Pourahmadi and Wang [1] for generating a random $p \times p$ correlation matrix. Briefly, the idea is to represent the correlation matrix using Cholesky factorization and $p(p-1)/2$ hyperspherical coordinates (i.e., angles), sample the angles from a particular distribution and then convert to the standard correlation matrix form. The angles are sampled from a distribution with probability density function $\sin^k(\theta)$ ($0 < \theta < \pi$, $k \geq 1$) using the efficient sampling algorithm described in [2].

Note

To cite this package please reference:

Makalic, E. & Schmidt, D. F. An efficient algorithm for sampling from $\sin^k(x)$ for generating random correlation matrices arXiv:1809.05212, 2018 <https://arxiv.org/abs/1809.05212>

A MATLAB-compatible implementation of the sampler in this package can be obtained from:

<https://au.mathworks.com/matlabcentral/fileexchange/68810-randcorr>

References

[1] Mohsen Pourahmadi and Xiao Wang, Distribution of random correlation matrices: Hyperspherical parameterization of the Cholesky factor, Statistics & Probability Letters, Volume 106, Pages 5-12, 2015.

[2] Enes Makalic and Daniel F. Schmidt An efficient algorithm for sampling from $\sin^k(x)$ for generating random correlation matrices, arXiv:1809.05212, 2018.

See Also

[randcorr.sample.sink](#)

Examples

```
# -----
# Example 1: Generate a 5x5 correlation matrix
C = randcorr(5)

# Example 2: Generate a 1000x1000 correlation matrix
C = randcorr(1000)
```

randcorr.sample.sink *Sample from the (unnormalized) distribution $\sin(x)^k$, $0 < x < \pi$, $k \geq 1$*

Description

Sample from the (unnormalized) distribution $\sin(x)^k$, $0 < x < \pi$, $k \geq 1$

Usage

```
randcorr.sample.sink(k)
```

Arguments

k The k parameter of the distribution. If this is a vector, the function draws a random variate for every entry in k .

Value

A vector of samples with length equal to the length of k

Details

This code generates samples from the $\sin(x)^k$ distribution using the specified vector k .

References

Enes Makalic and Daniel F. Schmidt An efficient algorithm for sampling from $\sin^k(x)$ for generating random correlation matrices, arXiv:1809.05212, 2018.

See Also

[randcorr](#)

Examples

```
# -----  
# Example 1: Draw a random variate from  $\sin(x)$ ,  $0 < x < \pi$   
x = randcorr.sample.sink(1)  
  
# Example 2: Draw a million random variate from  $\sin^3(x)$ ,  $0 < x < \pi$   
x = randcorr.sample.sink( matrix(3, 1e6,1) )  
mean(x)  
var(x)
```

Index

- * **correlation**

- randcorr-package, [2](#)

- * **distribution**

- randcorr-package, [2](#)

- * **matrix,**

- randcorr-package, [2](#)

randcorr, [2](#), [3](#), [4](#)

randcorr-package, [2](#)

randcorr.sample.sink, [2](#), [3](#), [4](#)