Package ‘NORTARA’

February 19, 2015

Title  Generation of Multivariate Data with Arbitrary Marginals
Version  1.0.0
Description  An implementation of a specific method for generating
n-dimensional random vectors with given marginal distributions and
correlation matrix. The method uses the NORTA (NORmal To Anything)
approach which generates a standard normal random vector and then
transforms it into a random vector with specified marginal distributions
and the RA (Retrospective Approximation) algorithm which is a generic
stochastic root-finding algorithm. The marginals can be continuous or
discrete. See the vignette of package for more details.

URL  https://github.com/superdesolator/NORTARA
BugReports  https://github.com/superdesolator/NORTARA/issues
Depends  R (>= 3.1.2)
License  MIT + file LICENSE
LazyData  true
Imports  corpcor(>= 1.6.7), Matrix(>= 1.1-4)
Suggests  knitr
VignetteBuilder  knitr
Author  Po Su [aut, cre]
Maintainer  Po Su <desolator@sjtu.edu.cn>
NeedsCompilation  no
Repository  CRAN
Date/Publication  2014-12-20 15:46:33

R topics documented:

  nortaRA-package .................................................. 2
  BoundingRA ....................................................... 2
  check_input_cormat ............................................. 4
  genNORTARA ....................................................... 5
  valid_input_cormat .............................................. 8
Description

This package implements a specific method for generating n-dimensional random vectors with given marginal distributions and correlation matrix. The method uses the NORTA(NORmal To Anything) approach which generates a standard normal random vector and then transforms it into a random vector with specified marginal distributions and the RA(Retrospective Approximation) algorithm which is a generic stochastic root-finding algorithm. Data generation is accomplished by first using `BoundingRA` to calculate an intermediate multivariate normal correlation matrix, then the matrix is used to generate samples from multivariate normal distribution. The engine function `gennortara` will transforms the normal samples to the wanted data set with specified inputmarginals from users. The function `valid_input_cormat` returns the lower and upper bounds of the mixture pre-specified marginal distributions. The function `check_input_cormat` checks the input target correlation matrix whether it is in the lower and upper bounds, if in the bounds, then the function will return `TRUE` it means the input target correlation matrix is feasible, otherwise, it will print the elements’ positions which are out of bounds and give an error message.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>nortaRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.0.0</td>
</tr>
<tr>
<td>Date</td>
<td>2014-12-06</td>
</tr>
<tr>
<td>License</td>
<td>MIT + file LICENSE</td>
</tr>
</tbody>
</table>

Author(s)

Po Su
Maintainer: Po Su <desolator@sjtu.edu.cn>

Description

The function computes an intermediate multivariate correlation matrix with a specific RA(Retrospective Approximation) algorithm called bounding RA.
BoundingRA

Usage

BoundingRA(cor_matrix, invcdfnames, paramslists, m1 = 60, c1 = 2, c2 = 1,
delta1 = 1e-04, sigma0 = 0.01, epsilon = 1e+00, maxit = 1000)

Arguments

cor_matrix The specified input correlation matrix.
invcdfnames A character sequence of the marginals' inverse cdf names.
paramslists A list contains lists of params of the marginals as the same order as invcdfnames.
m1 The initial sample size.
c1 The sample-size multiplier(c1>1).
c2 The step-size multiplier(c2>0).
delta1 The initial step size(deltla1>0).
sigma0 The standard error tolerance.
epsilon The initial error tolerance.
maxit The maximum number of numerical searches.

Details

The function computes an intermediate multivariate correlation matrix with a specific RA(Retrospective Approximation) algorithm called bounding RA. Then the result matrix will be used to applying the NORTA approach to get \( n \) observations which have specified marginals and target correlation matrix cor_matrix.

Value

An intermediate normal correlation matrix of the same size as cor_matrix.

Note

In this implementation of the NORTA approach with bounding RA algorithm according to the reference paper, the initial sample size is adjusted from 40 to 60, the other parameters remain the same. And the third choice of random seeds w_bar is used for efficiency while the Appendix of the paper uses the first choice.

Author(s)

Po Su

References


See Also
genNORTARA, valid_input_cormat, check_input_cormat
check_input_cormat

Test the input correlation matrix whether it is in the feasible bounds.

Description

The function uses the lower and upper correlation bounds from results of the function valid_input_cormat to test the users’ input correlation matrix.

Usage

check_input_cormat(invcdfnames, paramslists, cor_matrix)

Arguments

invcdfnames A character sequence of the marginals’ inverse cdf names.
paramslists A list contains lists of params of the marginals as the same order as invcdfnames.
cor_matrix The input correlation matrix to be checked.
Details

The function uses the lower and upper correlation bounds from results of the function valid_input_cormat to test the users' input correlation matrix. If all the elements in the input correlation matrix are in the bounds, the function will return `TRUE`, otherwise it will print out the elements' positions in the input correlation matrix which are out of the lower and upper bounds.

Value

If all the elements of the input correlation matrix all in the bounds the function will return `TRUE`, otherwise it will print messages about the out of bounds elements' positions and then give an error message to users.

Note

Because of the random samples, the results of the function may be a little different each time.

See Also

`BoundingRA, valid_input_cormat, genNORTARA`

Examples

```r
## not run:
invcdfnames <- c("qt","pois","qnorm")
paramslists <- list(
  m1 = list(df = 3),
  m2 = list(lambda = 5),
  m3 = list(mean = 0, sd = 1)
)
cor_matrix_correct <- matrix(c(1,0.5,-0.3,0.5,1,0.4,-0.3,0.4,1), 3)
cor_matrix_wrong <- matrix(c(1,0.94,-0.3,0.94,1,0.4,-0.3,0.4,1), 3)
check_input_cormat(invcdfnames, paramslists, cor_matrix_correct)
check_input_cormat(invcdfnames, paramslists, cor_matrix_wrong)
## end(not run)
```

The function simulates a data set with specified input correlation matrix `cor_matrix` and pre-specified marginals `invcdfnames` using bounding RA and NORTA methods.
Usage

genNORTARA(n, cor_matrix, invcdfnames, paramslists = NULL, 
defaultindex = NULL, m1 = 60, c1 = 2, c2 = 1, delta1 = 1e-04, 
sigma0 = 0.01, epsilon = 1e+00, maxit = 1000)

Arguments

n
Number of observations.

cor_matrix
specified input correlation matrix.

invcdfnames
A character sequence of the marginals’ inverse cdf(cumulative distribution function) names.

paramslists
A list contains lists of params of the marginals excluded the index(es) in defaultindex meanwhile as the same order as invcdfnames, the names of the arguments of the inner lists should keep the same with the function arguments matching rules with the arguments of invcdfnames functions.

defaultindex
The index number sequence which indicates the corresponding inverse cdfs use the default argument values.

m1
The initial sample size.

c1
The sample-size multiplier(c1>1).

c2
The step-size multiplier(c2>0).

delta1
The initial step size(deltal>0).

sigma0
The standard error tolerance.

epsilon
The initial error tolerance.

maxit
The maximum number of numerical searches.

Details

The function simulates a date set with variables from arbitrary(continuous or discrete) marginal distributions which have a correlation matrix cor_matrix. The pre-specified marginals are described by invcdfnames,paramslists,defaultindex, the later two arguments will be combined into a full paramslists which has the same length as invcdfnames. The function uses result of the function BoundingRA which is an implementation of a specific RA(Retrospective Approximation) algorithm called bounding RA. With the result, the function uses the NORTA(NORmal To Anything) approach which generates a standard normal random vector and then transforms it into a random vector with specified marginal to generates the wanted samples.

Value

A matrix of size n * (ncol(cor_matrix)) from pre-specified marginals which also have an asymptotically correlation matrix to specified input correlation matrix cor_matrix.
Note

1. The inverse cdf functions should have its first argument as a vector.
2. The inverse cdf functions like qweibull should not use default values because the shape argument does not have a default value. So we should not put its index in invcdfnames into defaultindex.
3. The function won’t accept the invcdfnames all be 'qnorm', you can generate multi-normal variables by other packages. The function will give you a error message, but if you use a = qnorm, then using new name, the function won’t find it, just be careful.
4. You may get a warning message indicate the Nearest positive definite matrix is used. It happens when your inputs won’t generate a positive definite intermediate normal correlation matrix. And in this case, the cor(res) may not very close to cor_matrix.

Author(s)

Po Su

References


See Also

BoundingRA, valid_input_cormat, check_input_cormat

Examples

```r
## not run:
invcdfnames < c("qt","qpois","qnorm","qweibull","qunif")
# The following usage :
# a <- qt; b <- qnorm; f <- stats::qweibull (It is also the way you can use functions
# from other packages)
# invcdfnames <- c("a","qpois","b","f","qunif") will also be ok!
paramslists <- list(
  m1 = list(df = 3),
  m2 = list(lambda = 5),
  m4 = list(shape = 1)
)
defaultindex <- c(3,5)
# It means the 3rd and 5th invcdf should use its default arguments.
# That means qnorm using mean = 0, sd = 1, qunif using min =0 ,max =1 and so on.
cor_matrix <- matrix(c(1,0,0.4,0.1,0.7,-0.2,-0.4,
  1,0,0.4,0.9,0.1,0.4,1.0,
  0.5,0.5,0.7,0.4,0.5,1.0,
  0.7,-0.2,0.9,0.5,0.7,1.0),5,5)
res <- genNORTARA(10000,cor_matrix,invcdfnames,paramslists,defaultindex)
# May get warning message indicating nearest positive definite is used,It's
# normal but the cor(res) may not very close to cor_matrix.
cor(res)
```
valid_input_cormat

Computes the lower and upper correlation bounds for the input marginals.

Description

The function computes the lower and upper correlation bounds for the input marginals.

Usage

valid_input_cormat(invcdfnames, paramslists)

Arguments

invcdfnames  A character sequence of the marginals' inverse cdf names.
paramslists  A list contains lists of params of the marginals as the same order as invcdfnames.

Details

The function computes the lower and upper correlation bounds for the input marginals. And returns a list of lower and upper correlation matrices for the target correlations based on the marginals, the matrices' dimensions are decided by the length of invcdfnames.

Value

A list of two matrices. The min_valid_cormat contains the lower bounds and the max_valid_cormat contains the upper bounds of the feasible correlations.

Note

Because of the random samples, the results of the function may be a little different each time.

References

valid_input_cormat

See Also

BoundingRA, check_input_cormat, genNORTARA

Examples

```r
## Not run:
invcdfnames <- c("qt","pois","qnorm")
paramslists <- list(
  m1 = list(df = 3),
  m2 = list(lambda = 5),
  m3 = list(mean = 0, sd = 1)
)
valid_input_cormat(invcdfnames, paramslists)

## End(Not run)
```
Index

BoundingRA, 2, 5, 7, 9
check_input_cormat, 2, 3, 4, 7, 9
genNORTARA, 2, 3, 5, 5, 9
nortaRA (nortaRA-package), 2
nortaRA-package, 2
valid_input_cormat, 2, 3, 5, 7, 8