Package ‘mvLognCorrEst’

March 15, 2023

Type Package

Title Sampling from a multivariate Log-Normal distribution using Log-Normal parameters and Estimation of Indirect Correlations

Version 1.0.0

Author Alessandro De Carlo

Maintainer Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

Description This package contains a set of function that allows sampling from a multivariate Log-Normal distribution using the parameters (means,sd,correlation matrix) of Log-Normal variables with a not-diagonal covariance matrix. It can also estimate unknown (indirect) correlations from an uncomplete correlation structure.

License GPL

Imports Matrix, utils, stats, qgraph, igraph (>= 1.2.7), pracma, NlcOptim

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

R topics documented:

`mvLognCorrEst` *Introduction to package usage*

This R package was developed to support simulation-based analysis for scenarios that require to sample from a multivariate Log-Normal distribution with a not-diagonal covariance matrix and to estimate unknown ('indirect') correlations between a set of correlated variables. These tasks can be performed either independently or sequentially using the functions of `mvLognCorrEst` as outlined by the flowchart in Fig.1.

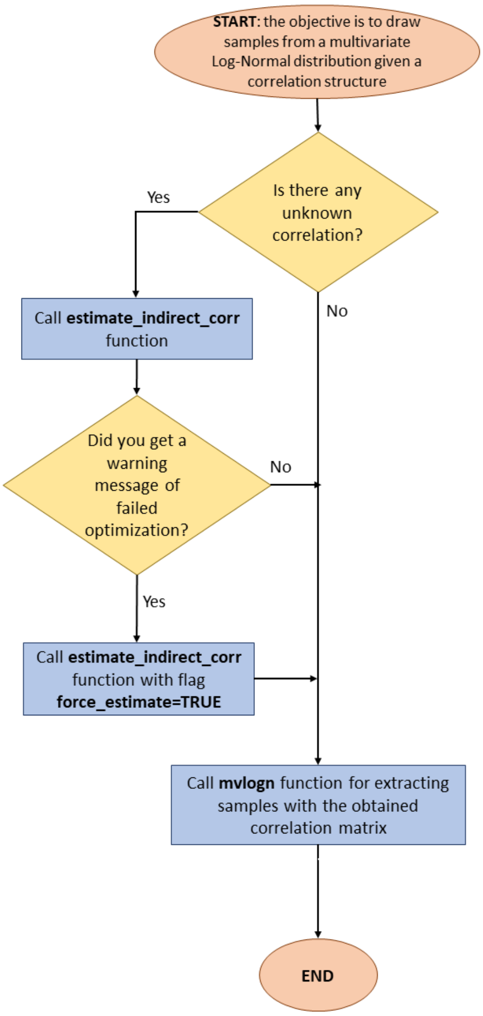

Figure 1: Flow chart of the typical pipeline that can be implemented by the `mvLognCorrEst` package.

`estimate_corr_bounds` *Estimation of bounds of indirect correlations*

Description

This function returns the ranges of indirect correlations to be estimated. `estimate_corr_bounds` receives in input a correlation matrix where indirect correlations to be estimated must be identified with `NA` values. From input correlation structure the associated graph is derived. For each indirect correlation between a generic couple of variables ($X1$,$X2$), all the possible paths in the graph that links $X1$ and $X2$ visiting one node at most once are considered.

The cost of each path is computed by multiplying the correlations met along the path and, the minimum and the maximum costs are used to identify the indirect correlation range. If there is not any path between two nodes, maximum and minimum values will be set to `NA`. If there is not any path between two nodes, maximum and minimum values will be set to `NA`. If there is a unique path, the bound is computed considering $cost + / - widen\_factor * cost$, where $cost$ is the cost of the unique existing path. The default value of $widen\_factor$ is 0.2.

Usage

```
estimate_corr_bounds(corrMat,widen_factor=0.2)
```

Arguments

`corrMat`	Matrix Object which represents a correlation matrix. Indirect correlations to be estimated must be indicated by `NA`. This matrix must be symmetric, thus it must contain at least two `NA` values.
`widen_factor`	number between 0 and 1. If there is a unique path, the range for that indirect correlation is computed considering $cost + / - widen\_factor * cost$, where $cost$ is the cost of the unique existing path. Default value is 0.2.

Value

A matrix object with N(= number of indirect correlations) rows and 4 columns reporting:

- `var1`: numerical index of $X1$, the first variable of the couple for which the indirect correlation has to be estimated
- `var2`: numerical index of $X2$, the second variable of the couple for which the indirect correlation has to be estimated
- `lower`: lower bound for the range of indirect correlation between `var1` and `var2`
- `upper`: upper bound for the range of indirect correlation between `var1` and `var2`.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

See Also

`graph_from_adjacency_matrix`

`all_simple_paths`

Examples

```
# create a correlation matrix, define some correlations
c_start <- diag(rep(1,6))
c_start[1,2] <- -0.6
c_start[1,3] <- -0.75
```

```
c_start[2,3] <-0.95
c_start[2,4] <- 0.75
c_start[2,6] <- -0.6
c_start <- c_start+t(c_start)-diag(rep(1,ncol(c_start)))
#set to NA indirect correlations
c_start[c_start==0]<-NA
#plot correlation graph
plot_graph_corr(c_start,"Graph of Correlation Matrix")
#get bounds of correlations
estimate_corr_bounds(c_start)

#output: variable 5 is not directly correlated with any of the other variables so it
#is impossible to establish a path to it. Bounds of indirect correlation involving variable 5 are set to NA

#       var1 var2     lower   upper
#[1,]    1    4 -0.534375 -0.4500
#[2,]    1    5        NA      NA
#[3,]    1    6  0.360000  0.4275
#[4,]    2    5        NA      NA
#[5,]    3    4  0.337500  0.7125
#[6,]    3    5        NA      NA
#[7,]    3    6 -0.570000 -0.2700
#[8,]    4    5        NA      NA
#[9,]    4    6 -0.540000 -0.3600
#[10,]    5    6        NA      NA
```

---

estimate_indirect_corr

Estimation of Indirect Correlations

Description

This function estimates indirect correlations starting from the incomplete correlation matrix in input. Indirect correlations to be estimated must be indicated by NA values in the input correlation matrix.

Usage

```
estimate_indirect_corr(corrMatStart, force_estimate = FALSE, widen_factor=0.2)
```

Arguments

corrMatStart: Matrix object which represents a correlation matrix. Indirect correlations to be estimated must be indicated by NA. This matrix must be symmetric, thus it must contain at least two NA values.

force_estimate: Boolean flag. When this flag is *TRUE*, if the obtained correlation matrix is not positive definite, it is approximated to the nearest positive definite matrix based on the Frobenius norm. Matrix approximation may alter fixed initial correlations. If this flag is set to *FALSE* (default option), matrix approximation is skipped and a warning message is returned.

widen_factor: number between 0 and 1. If there is a unique path, the range for that indirect correlation is computed considering $cost +/- widen\_factor * cost$, where $cost$ is the cost of the unique existing path. Default value is 0.2.

Details

Indirect correlations are estimated solving a constrained optimization problem. Starting from the fixed correlations, a correlation graph is built. Then, for each couple of variables whose indirect correlation is unknown (i.e. `NA` values), all the possible paths among them are considered (without visiting a node more than once). The cost of each path is computed by multiplying the correlations along it. The maximum and the minimum costs provide a reasonable range for the indirect correlation value. If there is not any path between two nodes, that indirect correlation will not be estimated and it will be automatically set to 0. If there is a unique path, the range for that indirect correlation is computed considering $cost + / - widen\_factor * cost$, where $cost$ is the cost of the unique existing path. The default value of $widen\_factor$ is 0.2.

Given the bounds of indirect correlations, a constrained optimization problem is solved by minimizing the negative of minimum eigenvalue of correlation matrix. The starting values for the indirect correlation values are set equal to the middle-point of the computed bounds. If the estimated matrix is not positive definite, user can force a second optimization step in which the previously obtained matrix is approximated to its nearest positive definite matrix. Frobenius norm is used to measure distance between matrices. Note that this step may alter initial fixed correlations.

An indirect correlation between two variables can be estimated only if they are linked by at least one path in the correlation graph. If for all indirect correlations declared does not exist any path, this function prints a warning message and plots the correlation graph to support the debug.

Value

A list containing

`corrMatFinal` Matrix object containing the final correlation matrix with indirect correlations estimated.

`optim` List of objects containing the outputs provided by the solver (`fmincon` of `pracma` package) used for the constrained optimization. It is returned if the optimization step converges to a positive definite matrix or if the optimization step fails and `force_estimate` is set to `FALSE`.

`optim1` The same of `optim`. It is returned only when constrained optimization does not converge to a positive definite correlation matrix and `force_estimate` is set to `TRUE`.

`optim2` List of objects containing the outputs provided by the function `nearPD` of `Matrix` package used to approximate the matrix obtained by solving the constrained optimization problem with the nearest positive definite correlation matrix. It is returned only when constrained optimization does not converge to a positive definite correlation matrix and `force_estimate` is set to `TRUE`.

`optimizationBounds` A matrix object with N(= number of indirect correlations) rows and 4 columns reporting:

- `var1`: numerical index of $X1$, the first variable of the indirect correlation couple
- `var2`: numerical index of $X2$, the second variable of the indirect correlation couple
- `lower`: lower bound for the range of indirect correlation between `var1` and `var2`
- `upper`: upper bound for the range of indirect correlation between `var1` and `var2`.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

See Also

nearPD

fmincon

estimate_corr_bounds

Examples

```
#define initial correlation structure
c_start <- diag(rep(1,10))
c_start[1,2] <- -0.6
c_start[1,3] <- -0.75
c_start[2,3] <-0.95
c_start[2,4] <- 0.75
c_start[2,6] <- -0.6
c_start[2,8] <- 0.75
c_start[3,4] <- 0.6
c_start[3,8] <-0.75
c_start[4,7] <- 0.6
c_start[4,8]<-0.75
c_start[5,7] <- -0.95
#symmetric correlation structure
c_start <- c_start+t(c_start)-diag(rep(1,ncol(c_start)))
#assign NA to indirect correlations to be estimated
c_start[c_start==0]<-NA
#names of variables
colnames(c_start)<- paste(rep("X",10),1:10,sep = "")
rownames(c_start) <- paste(rep("X",10),1:10,sep = "")
# plot initial correlation matrix
plot_graph_corr(c_start,"Graph of Initial Correlation Matrix")
r<-estimate_indirect_corr(c_start)
#see final output
plot_graph_corr(r$corrMatFinal,'Graph of Final Correlation Matrix')
```

---

get_logNcorr_bounds   *Computation of the ranges of Correlations for Log-Normal distributions*

Description

Correlations between Log-Normal distributed variables should be in specific ranges depending on their CVs. These bounds represent a necessary condition for obtaining a positive definite covariance/correlation matrix for the underlying normal distribution [1].

Usage

```
get_logNcorr_bounds(mu, sd)
```

Arguments

mu	Array object which contains mean values of variables with a Log-Normal distribution.
sd	Array object which contains sd values of variables with a Log-Normal distribution.

Details

If $X = (x1, ..., xN)$ is a set of variables with a Log-Normal distribution with parameters $\theta, \Omega, Y = ln(X)$ will be a set of Normal parameters with parameters μ, Σ. The bond between X and Y, as well as between the parameters of the two multivariate distribution, is a non-linear transformation. Thus, Log-Normal correlation structure has to fulfill the bound described in [1] in order to obtain a positive definite covariance matrix for the underlying Normal distribution.

Value

A Matrix object with a number of rows equal to the number of Log-Normal variables couples and 4 columns containing:

- `var1`: numerical index of the first Log-Normal variable of the couple
- `var2`: numerical index of the second Log-Normal variable of the couple
- `lower`: lower bound for the range of Log-Normal correlation between `var1` and `var2`
- `upper`: upper bound for the range of Log-Normal correlation between `var1` and `var2`.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

References

[1] Henrique S. Xavier, Filipe B. Abdalla, Benjamin Joachimi, Improving lognormal models for cosmological fields, Monthly Notices of the Royal Astronomical Society, Volume 459, Issue 4, 11 July 2016, Pages 3693–3710, https://doi.org/10.1093/mnras/stw874

Examples

```
sd2 <- array(c(1,2,0.5,3)) #array with standard deviations of variables
mu2 <- array(c(3,0.2,1,8)) #array with means of variables
get_logNcorr_bounds(mu2,sd2)

#returns
#      var1 var2   lower     upper
# [1,]    1    2 -0.1506242 0.3025073
# [2,]    1    3 -0.8529277 0.9942674
# [3,]    1    4 -0.8885903 0.9996221
# [4,]    2    3 -0.1275056 0.3517665
# [5,]    2    4 -0.1443341 0.3146269
# [6,]    3    4 -0.8398526 0.9968250
```

`logn_to_normal` *Convert Log-Normal parameters to Normal parameters*

Description

This function is used to convert mean vector and covariance matrix of a multivariate Log-Normal distribution to mean vector and covariance matrix of the associated multivariate Normal distribution. Two conditions are evaluated before computing the parameters of Normal distribution:

1. For each couple of variables, $X1$, $X2$, it is tested if $\rho(X1, X2)$ satisfies the condition necessary to obtain a positive definite correlation/covariance matrix for the Normal distribution underlying the Log-Normal one. This condition is also sufficient for 2x2 matrix [1]. If this condition is not satisfied, a warning message is displayed but covariance matrix of the Normal distribution is still computed even if it is not positive definite.
2. For each couple of variables, $X1$, $X2$, it is tested if $\rho(X1, X2) * cv1 * cv2 > -1$. If this condition is not satisfied for all the couples the covariance matrix of the underlying Normal distribution cannot be computed. In fact, the covariance between $X1$, $X2$ of the Normal distribution, $\Sigma^2(X1, X2)$, is defined as $ln((\rho(X1, X2) * cv1 * cv2) + 1)$ [1]. In this case, a warning message will be displayed and a summary of the performed tests is returned.

Usage

```
logn_to_normal(mu, covMatrix)
```

Arguments

`mu`	Array object containing means of multivariate Log-Normal distribution.
`covMatrix`	Matrix object containing covariance matrix of multivariate Log-Normal distribution.

Value

A list containing:

`muN` Array object. It contains mean of the multivariate Normal distribution associated to Log-Normal one.

`muN` Array object. It contains mean of the multivariate Normal distribution associated to Log-Normal one.

`sigmaN` Matrix object. It contains covariance matrix of the multivariate Normal distribution associated to Log-Normal one.

`is_valid_logN_corrMat`
Boolean flag which indicates if the input correlation matrix satisfies condition 1.

`can_log_transf_covMat`
Boolean flag which indicates if the input correlation matrix satisfies condition 2.

`validation_res` Matrix object which contains a brief summary of the tested conditions on the input correlation matrix. It is composed by N(=number of Log-Normal variable couples) rows and 8 columns:

- `var1`: numerical index of the first Log-Normal variable of the couple, $X1$
- `var2`: numerical index of the second Log-Normal variable of the couple, $X2$

- lower: lower bound for the range of Log-Normal correlation between var1 and var2
- upper: upper bound for the range of Log-Normal correlation between var1 and var2
- tested_corr: input correlation value between var1 and var2
- is_valid_bound: numerical flag. If 1 tested_corr is inside the range, 0 otherwise
- tested_for_logTransf: value computed for testing condition 2 ($\rho(X1, X2) * cv1 * cv2$)
- can_logTransf: numerical flag. If 1 covariance of normal distribution between var1 and var2 can be computed, 0 otherwise.

Note

This function tries to compute the covariance matrix of the multivariate Normal distribution even if the Log-Normal covariance matrix is not positive definite. User should check that input matrix is positive definite, however in this situation logn_to_normal returns a warning message. Even if input matrix is positive definite and conditions 1 and 2 are fulfilled, covariance matrix of multivariate Normal distribution could not be positive definite because condition 1 is only necessary [1]. In order to fix this problem, nearPD of Matrix package could be used.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

References

[1] Henrique S. Xavier, Filipe B. Abdalla, Benjamin Joachimi, Improving lognormal models for cosmological fields, Monthly Notices of the Royal Astronomical Society, Volume 459, Issue 4, 11 July 2016, Pages 3693–3710, https://doi.org/10.1093/mnras/stw874

See Also

nearPD

get_logNcorr_bounds

validate_logN_corrMatrix

Examples

```
#define correlations
corr<- diag(rep(1,4))
corr[1,4] <- 0.9
corr[4,1]<-corr[1,4]
corr[2,4] <- -0.3
corr[4,2] <- corr[2,4]
corr[3,2] <- -0.2
corr[2,3] <- corr[3,2]

#define sd of variables
sd2 <- array(c(rep(1,4)))
#obtain covariance matrix
covMatrix2 <- sd2%*%t(sd2)*corr
#define mean vector
```

```
mu2 <- array(rep(2.5,4))
logn_to_normal(mu2,covMatrix2)

#output:

# $is_valid_logN_corrMat
# [1] TRUE

# $can_log_transf_covMat
# [1] TRUE
#
# $validation_res
#     var1 var2    lower upper tested_corr is_valid_bound tested_for_logTransf can_logTransf
# [1,]    1    2 -0.862069     1         0.0              1                0.000             1
# [2,]    1    3 -0.862069     1         0.0              1                0.000             1
# [3,]    1    4 -0.862069     1         0.9              1                0.144             1
# [4,]    2    3 -0.862069     1        -0.2              1               -0.032             1
# [5,]    2    4 -0.862069     1        -0.3              1               -0.048             1
# [6,]    3    4 -0.862069     1         0.0              1                0.000             1

# $muN
# [1] 0.8420807 0.8420807 0.8420807 0.8420807

# $sigmaN
# [,1]        [,2]        [,3]        [,4]
# [1,] 0.1484200  0.00000000  0.00000000  0.13453089
# [2,] 0.0000000  0.14842001 -0.03252319 -0.04919024
# [3,] 0.0000000 -0.03252319  0.14842001  0.00000000
# [4,] 0.1345309 -0.04919024  0.00000000  0.14842001
```

`mvlogn` *Sampling random numbers from a multivariate Log-Normal distribution*

Description

This function allows to generate random samples from a Log-Normal distribution using the parameters of Log-Normal distribution instead of the ones of underlying Normal. A schematic representation of the operations performed by the `mvlogn` is provided by the flowchart in Fig.2.

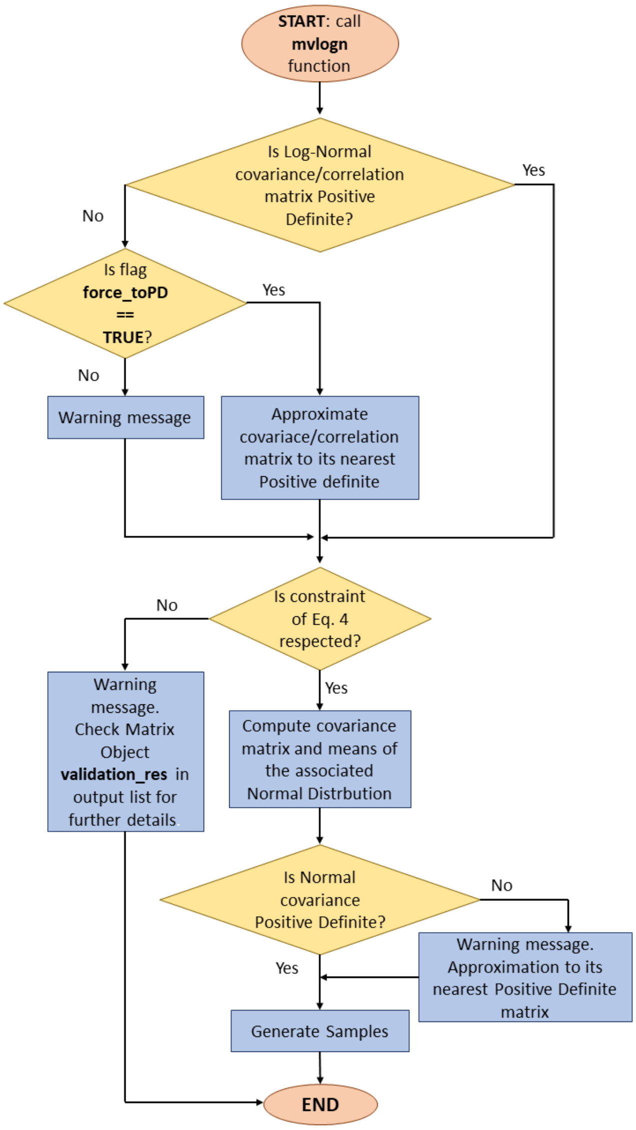

Figure 2: Flow chart which illustrates the operations performed by the `mvlogn` function.

Usage

```
mvlogn(Nsamples, ..., force_toPD = TRUE, full_output = TRUE)
```

Arguments

`Nsamples` Number of random samples to extract from the multivariate Log-Normal distribution.

`...` One of the following five combination of named arguments:

1. `mu=input_mean, covMatrix = input_covariance`
2. `mu=input_mean, sd=input_sd, corrMatrix=input_correlations`
3. `mu=input_mean, cv=input_cv, corrMatrix=input_correlations`
4. `sd=input_sd, cv=input_cv, corrMatrix=input_correlations`
5. `sd=input_sd, cv=input_cv, covMatrix = input_covariance`

where:

- `mu`: Array object containing means of the multivariate Log-Normal distribution
- `sd`: Array object containing standard deviations of the multivariate Log-Normal distribution
- `cv`: Array object containing CVs of the multivariate Log-Normal distribution. CVs are **not** expressed with percentage.
- `covMatrix`: Matrix object containing covariance matrix of the multivariate Log-Normal distribution
- `corrMatrix`: Matrix object containing correlation matrix of the multivariate Log-Normal distribution.

`force_toPD` Boolean flag. When it is set to *TRUE* (default value), if the input correlation/covariance matrix is not positive definite, it is approximated to its nearest positive definite. If it is set to *FALSE*, approximation to nearest positive definite is not performed and a warning message is printed.

`full_output` Boolean flag. If it is set to *TRUE* (default value), the covariance matrix of Normal distribution is included in the output list. If the covariance matrix of Normal distribution is not positive definite, output will contain the initial version of Normal covariance matrix, its nearest positive definite and the normalized Frobenius and Infinity norms of their difference.

Details

The correlation/covariance structure of Log-Normal variables must be specified. If this matrix is not positive definite, it will be approximated to its nearest positive definite using `nearPD` function of `Matrix` package. In this case, the function return a warning message. It is necessary that Normal covariance matrix is positive definite because this sampling strategy is based on Cholesky decomposition.

This function evaluates two conditions before computing the parameters of Normal distribution:

1. For each couple of variables, $X1$, $X2$, it is tested if $\rho(X1, X2)$ satisfies the condition necessary to obtain a positive definite correlation/covariance matrix for the Normal distribution underlying the Log-Normal one. This condition is also sufficient for 2x2 matrix [1]. If this condition is not satisfied, a warning message is displayed but covariance matrix of the Normal distribution is still computed even if it is not positive definite.
2. For each couple of variables, $X1$, $X2$, it is tested if $\rho(X1, X2) * cv1 * cv2 > -1$. If this condition is not satisfied for all the couples the covariance matrix of the underlying Normal distribution cannot be computed. In fact, the covariance between $X1$, $X2$ of the Normal distribution, $\Sigma^2(X1, X2)$, is defined as $ln((\rho(X1, X2) * cv1 * cv2) + 1)$ [1]. In this case, a warning message will be displayed and a summary of the performed tests is returned.

If condition 2 is fulfilled, independently by condition 1, sampling is guaranteed. Sampling strategy is based on the following steps:

1. Generation of random uniform matrix `NsamplesxNvariables`.
2. Computation of the inverse CDF of standard Normal distribution for these samples.
3. Multiplication of the matrix obtained in previous step by Cholesky decomposition of Normal covariance matrix.
4. Sum of the mean values of each variable.
5. Exponential transformation.

The approximation to the nearest positive definite of initial covariance/correlation matrix and/or Normal covariance will alter correlation/covariance structure among variables. In order to appreciate this difference, this function returns both Frobenius and Infinity norms. See the section below for further details.

Value

A list containing:

- `samples` Matrix object. Each row represents a sample. Thus, the number of rows is equal to the number of samples `Nsamples`, while the number of columns is equal to the number of Log-Normal variables.
- `input_covMat_adj` Matrix object with the nearest positive definite matrix to the input covariance matrix. This object is returned only if input covariance matrix is not positive definite and `flag_force=TRUE`.
- `input_corrMat_adj` Matrix object with the nearest positive definite matrix to the input correlation matrix. This object is returned only if input covariance matrix is not positive definite and `flag_force=TRUE`.
- `normF_input_adj` Scalar double. It represents the Frobenius Norm of the difference between input covariace/correlation matrix and its nearest positive definite, normalized for the Frobenius Norm of input matrix. This object is returned only if input covariance matrix is not positive definite and `flag_force=TRUE`.
- `normInf_input_adj` Scalar double. It represents the Infinity norm of the difference between input covariace/correlation matrix and its nearest positive definite, normalized for the Infinity norm of input matrix. This object is returned only if input covariance matrix is not positive definite and `flag_force=TRUE`.
- `validation_res` Matrix object which contains a brief summary of the tested conditions on the input correlation matrix. It is composed by N(=number of Log-Normal variable couples) rows and 8 columns:
  - `var1`: numerical index of the first Log-Normal variable of the couple, $X1$
  - `var2`: numerical index of the second Log-Normal variable of the couple, $X2$
  - `lower`: lower bound for the range of Log-Normal correlation between `var1` and `var2`
  - `upper`: upper bound for the range of Log-Normal correlation between `var1` and `var2`
  - `tested_corr`: input correlation value between `var1` and `var2`
  - `is_valid_bound`: numerical flag. If 1 `tested_corr` is inside the range, 0 otherwise
  - `tested_for_logTransf`: value computed for testing condition 2 ($\rho(X1, X2) * cv1 * cv2$)
  - `can_logTransf`: numerical flag. If 1 covariance of normal distribution between `var1` and `var2` can be computed, 0 otherwise.

`normal_cov` Matrix object. It contains covariance matrix of the underlying multivariate Normal distribution associated to Log-Normal one when Normal covariance is not approximated to its nearest positive definite. This object is returned only if `full_output=TRUE`.

normal_cov_not_adjusted Matrix object. It contains covariance matrix of multivariate Normal distribution associated to Log-Normal one when it is not positive definite. This object is returned only if full_output=TRUE and Normal covariance matrix is approximated to its nearest positive definite.

normal_cov_adjusted Matrix object. It contains the nearest positive definite matrix to the covariance matrix of the underlying multivariate Normal distribution. This object is returned only if full_output=TRUE and Normal covariance matrix is approximated to its nearest positive definite.

normF_normal_adj Scalar double. It represents the Frobenius norm of the difference between Normal covariance matrix and its nearest positive definite, normalized for the norm of the first. This value is returned only if full_output=TRUE and Normal covariance matrix is approximated to its nearest positive definite.

normInf_normal_adj Scalar double. It represents the Infinity norm of the difference between Normal covariance matrix and its nearest positive definite, normalized for the norm of the first. This value is returned only if full_output=TRUE and Normal covariance matrix is approximated to its nearest positive definite.

logN_corr_initial Matrix object. It contains the initial correlation structure of the multivariate Log-Normal distribution. This object is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

logNcorr_adj Matrix object. It contains the correlation structure of the multivariate Log-Normal distribution considering the approximation of Normal covariance matrix to its nearest positive definite. This object is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

normF_logN_corrMat Scalar double. It represents the Frobenius norm of the difference between initial correlation matrix and the one obtained after the approximation of Normal covariance matrix, normalized for the norm of the first. It is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

normInf_logN_corrMat Scalar double. It represents the Infinity norm of the difference between initial correlation matrix and the one obtained after the approximation of Normal covariance matrix, normalized for the norm of the first. It is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

logNcov_initial Matrix object. It contains the initial covariance matrix of multivariate Log-Normal distribution. This object is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

logNcov_adj Matrix object. It contains the covariance matrix of multivariate Log-Normal distribution considering the approximation of Normal covariance matrix to its nearest positive definite. This object is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

normF_logN_covMat Scalar double. It represents the Frobenius norm of the difference between initial covariance matrix and the one obtained after the approximation of Normal covariance matrix, normalized for the norm of the first. It is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

normF_logN_covMat Scalar double. It represents the Infinity norm of the difference between initial covariance matrix and the one obtained after the approximation of Normal covariance matrix, normalized for the norm of the first. It is returned only if the Normal covariance matrix is approximated to its nearest positive definite.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

References

[1] Henrique S. Xavier, Filipe B. Abdalla, Benjamin Joachimi, Improving lognormal models for cosmological fields, Monthly Notices of the Royal Astronomical Society, Volume 459, Issue 4, 11 July 2016, Pages 3693–3710, https://doi.org/10.1093/mnras/stw874

See Also

get_logNcorr_bounds

validate_logN_corrMatrix

logn_to_normal

nearPD

norm

Examples

```
#different ways to run this function
#define correlations
corr<- diag(rep(1,4))
corr[1,4] <- 0.9
corr[4,1]<-corr[1,4]
corr[2,4] <- -0.3
corr[4,2] <- corr[2,4]
corr[3,2] <- -0.2
corr[2,3] <- corr[3,2]
#define sd of variables
sd2 <- array(c(rep(1,4)))
#obtain covariance matrix
covMatrix2 <- sd2%*%t(sd2)*corr
#define mean vector
mu2 <- array(rep(2.5,4))
cv2 <- sd2/mu2
Nsamples <- 1000
#call 1
mvlogn(Nsamples,mu=mu2,covMatrix=covMatrix2)
#call 2
mvlogn(Nsamples,mu=mu2, sd=sd2, corrMatrix=corr)
#call 3
mvlogn(Nsamples, mu=mu2, cv=cv2, corrMatrix=corr)
# call 4
mvlogn(Nsamples, sd = sd2, cv=cv2, corrMatrix=corr)
#call 5
mvlogn(Nsamples, sd=sd2, cv=cv2, covMatrix=covMatrix2)

#all 5 calls allow to sample from the same distribution
```

---

normal_to_logn *Convert Normal parameters to Log-Normal parameters*

Description

This function converts mean vector and covariance matrix of a multivariate Normal distribution to mean vector and covariance matrix of the associated multivariate Log-Normal distribution.

Usage

```
normal_to_logn(mu, covMatrix)
```

Arguments

mu	Array object containing means of the multivariate Normal distribution.
covMatrix	Matrix object containing covariance matrix of the multivariate Normal distribution.

Value

A list containing:

muLn	Array object. It contains mean of the multivariate Log-Normal distribution associated to the Normal one.
sigmaLn	Matrix object. It contains covariance matrix of the multivariate Log-Normal distribution associated to the Normal one.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

See Also

logn_to_normal

Examples

```
#define correlations
corr<- diag(rep(1,4))
corr[1,4] <- 0.9
corr[4,1]<-corr[1,4]
corr[2,4] <- -0.3
corr[4,2] <- corr[2,4]
corr[3,2] <- -0.2
corr[2,3] <- corr[3,2]
#define sd of variables
sd2 <- array(c(rep(1,4)))
#obtain covariance matrix
covMatrix2 <- sd2%*%t(sd2)*corr
#define mean vector
mu2 <- array(rep(2.5,4))
normal_to_logn(mu2,covMatrix2)
#output:

# $muLn
# [1] 20.08554 20.08554 20.08554 20.08554
#
# $sigmaLn
#           [,1]    [,2]      [,3]      [,4]
# [1,] 693.2044    0.00000   0.00000  588.8459
# [2,]   0.0000  693.20436 -73.12923 -104.5614
# [3,]   0.0000  -73.12923 693.20436    0.0000
# [4,] 588.8459 -104.56139   0.00000  693.2044
```

`plot_graph_corr` *Plot of correlation graph*

Description

Function that generates a plot of the graph associated to a given correlation matrix.

Usage

```
plot_graph_corr(
  corrMat,
  title = "Graph of Correlation Matrix",
  title_size = 1.2,
  edge_width = 0.2,
  edge_label_size = 1.2,
  edge_label_position = 0.6,
  custom_labels = NULL,
  bg_color = "#F8F8F8",
  shade_factor = 0.2
)
```

Arguments

`corrMat`	Matrix object which represents correlation matrix.
`title`	String object which contains the title of the plot. Default values is "Graph of correlation matrix".
`title_size`	Double with the size of the title. Default value is 1.2.
`edge_width`	Double with the width of the edges. Default value is 0.2.
`edge_label_size`	Double with the size of the labels plotted on the edges. They correspond to the correlations. Default value is 1.2.
`edge_label_position`	Double in [0,1] interval which indicates the relative position of the label on the graph. Default value is 0.6.
`custom_labels`	Vector with custom node names. If it is *NULL*, column names of correlation matrix. are used. If correlation matrix has not column names, indices of column are taken. Default value is *NULL*.
`bg_color`	String with hexadecimal code of background color. Default value is gray "#F8F8F8".
`shade_factor`	Double for tuning edge shades. Default value is 0.2.

Note

No more than 3 letters can be showed in each node.

Author(s)

Alessandro De Carlo <alessandro.decarlo01@universitadipavia.it>

Examples

```
c_start <- diag(rep(1,10))
c_start[1,2] <- -0.6
c_start[1,3] <- -0.75
c_start[2,3] <-0.95
c_start[2,4] <- 0.75
c_start[2,6] <- -0.6
c_start[2,8] <- 0.75
c_start[3,4] <- 0.6
c_start[3,8] <-0.75
c_start[4,7] <- 0.6
c_start[4,8]<-0.75
c_start[5,7] <- -0.95
c_start <- c_start+t(c_start)-diag(rep(1,ncol(c_start)))
plot_graph_corr(c_start,"Graph of Correlation Matrix")
```

`validate_logN_corrMatrix` *Validation of Correlation matrix of Log-Normal distribution*

Description

Given a set of Log-Normal variables with their means, standard deviations and correlation matrix, this function evaluates if the correlation structure respects two conditions:

1. For each couple of variables, $X1$, $X2$, it is tested if $\rho(X1, X2)$ satisfies the condition necessary to obtain a positive definite correlation/covariance matrix for the Normal distribution underlying the Log-Normal one. This condition is also sufficient for 2x2 matrix [1]. If this condition is not satisfied, a warning message is displayed but covariance matrix of the Normal distribution is still computed even if it is not positive definite.
2. For each couple of variables, $X1$, $X2$, it is tested if $\rho(X1, X2) * cv1 * cv2 > -1$. If this condition is not satisfied for all the couples the covariance matrix of the underlying Normal distribution cannot be computed. In fact, the covariance between $X1$, $X2$ of the Normal distribution, $\Sigma^2(X1, X2)$, is defined as $ln((\rho(X1, X2) * cv1 * cv2) + 1)$ [1]. In this case, a warning message will be displayed and a summary of the performed tests is returned.

Usage

```
validate_logN_corrMatrix(mu, sd, corrMatrix)
```

Arguments

`mu`	Array object which contains mean values of variables with a Log-Normal distribution.
`sd`	Array object which contains sd values of variables with a Log-Normal distribution.
`corrMatrix`	Matrix object which contains correlations of variables with a Log-Normal distribution.

Value

A list containing:

`is_valid_logN_corrMat`
: Boolean flag which indicates if the input correlation matrix satisfies condition 1.

`can_log_transf_covMat`
: Boolean flag which indicates if the input correlation matrix satisfies condition 2.

`validation_res`
: Matrix object which contains a brief summary of the tested conditions on the input correlation matrix. It is composed by N(=number of Log-Normal variables couples) rows and 8 columns:

- `var1`: numerical index of the first Log-Normal variable of the couple, $X1$
- `var2`: numerical index of the second Log-Normal variable of the couple, $X2$
- `lower`: lower bound for the range of Log-Normal correlation between `var1` and `var2`
- `upper`: upper bound for the range of Log-Normal correlation between `var1` and `var2`
- `tested_corr`: input correlation value between `var1` and `var2`
- `is_valid_bound`: numerical flag. If 1 `tested_corr` is inside the range, 0 otherwise
- `tested_for_logTransf`: value computed for testing condition 2 ($\rho(X1, X2) * cv1 * cv2$)
- `can_logTransf`: numerical flag. If 1 covariance of normal distribution between `var1` and `var2` can be computed, 0 otherwise.

Note

Within this package, function used for sampling from the multivariate Log-Normal distribution (`mvlogn`) can overcome situations in which covariance matrix of the Normal distribution underlying the Log-Normal one is not positive definite (i.e. condition 1 not satisfied) by approximating it to the nearest positive definite. This may alter the desired correlation structure, but simulation is still allowed. On the contrary, if condition 2 is not satisfied, simulation cannot be performed because computing Normal covariance matrix is impossible to calculate since the argument of logarithmic transformation would be negative for some couples of variables.

Author(s)

Alessandro De Carlo `<alessandro.decarlo01@universitadipavia.it>`

References

[1] Henrique S. Xavier, Filipe B. Abdalla, Benjamin Joachimi, Improving lognormal models for cosmological fields, Monthly Notices of the Royal Astronomical Society, Volume 459, Issue 4, 11 July 2016, Pages 3693–3710, https://doi.org/10.1093/mnras/stw874

See Also

`nearPD`

`get_logNcorr_bounds`

`logn_to_normal`

Examples

```
#CONDITION 1 AND 2 SATISFIED
#input correlation matrix
corr<- diag(rep(1,4))
corr[1,4] <- 0.9
corr[4,1]<-corr[1,4]
corr[2,4] <- -0.2
corr[4,2] <- corr[2,4]
corr[3,2] <- -0.1
corr[2,3] <- corr[3,2]
#input standard deviations
sd2 <- array(c(rep(1,4)))
#input means
mu2 <- array(rep(2.5,4))
validate_logN_corrMatrix(mu2,sd2,corr)

#output

#$is_valid_logN_corrMat
#[1] TRUE
#
#$can_log_transf_covMat
#[1] TRUE

#$validation_res
#    var1 var2 lower    upper tested_corr  is_valid_bound tested_for_logTransf  can_logTransf
#[1,]    1    2 -0.862069    1        0.0            1                0.000            1
#[2,]    1    3 -0.862069    1        0.0            1                0.000            1
#[3,]    1    4 -0.862069    1        0.9            1                0.144            1
#[4,]    2    3 -0.862069    1       -0.1            1               -0.016            1
#[5,]    2    4 -0.862069    1       -0.2            1               -0.032            1
#[6,]    3    4 -0.862069    1        0.0            1                0.000            1
```