# Package ‘BinNor’

April 21, 2020

**Type** Package

**Title** Simultaneous Generation of Multivariate Binary and Normal Variates

**Version** 2.3.2

**Date** 2020-04-20

**Author** Anup Amatya, Hakan Demirtas, Ran Gao

**Maintainer** Ran Gao <rgao8@uic.edu>

**Depends** mvtnorm, corpcor, psych, Matrix

**Description** Generating multiple binary and normal variables simultaneously given marginal characteristics and association structure based on the methodology proposed by Demirtas and Doganay (2012) <DOI:10.1080/10543406.2010.521874>.

**License** GPL-2

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-04-21 04:10:02 UTC

## R topics documented:

BinNor-package *A package for simultaneous generation of binary and normal data.*

## Description

Provides R functions for generating multiple binary and normal variables simultaneously given the marginal characteristics and association structure via combining well established results from the random number generation literature, based on the methodology proposed by Demirtas and Doganay (2012).

## Details

| | |
|---|---|
| Package: | BinNor |
| Type: | Package |
| Version: | 2.3.2 |
| Date: | 2020-04-20 |
| License: | GPL-2 |
| LazyLoad: | yes |

There are eight functions in this package. The functions `lower.tri.to.corr.mat`, `validation.bin`, `validation.nor`, `validation.range` and `validation.nor` are designed to prevent obvious specification errors and to validate the specified quantities. The most important functions are `compute.sigma.star`, `jointly.generate.binary.normal` and `simulation`. The function `compute.sigma.star` computes the matrix of tetrachoric correlations that will be used in the generation of multivariate normal data whose some components are dichotomized to obtain binary variables. The function `jointly.generate.binary.normal` generates mixed data, and the function `simulation` is capable of repating this process many times and produces averages of some key statistical quantities across replications.

## Author(s)

Anup Amatya, Hakan Demirtas, Ran Gao

Maintainer: Ran Gao <rgao8@uic.edu>

## References

Demirtas, H., Doganay, B. (2012). Simultaneous generation of binary and normal data with specified marginal and association structures. Journal of Biopharmaceutical Statistics; 22(2), 223-236.

Demirtas, H., Amatya, A., and Doganay, B. (2014). BinNor: An R package for con-current generation of binary and normal data. Communications in Statistic-Simulation and Computation; 43(3), 569-579.

`compute.sigma.star` *Computes intermediate correlation matrix*

## Description

This function computes the intermediate correlation matrix by assembling tetrachoric correlations for binary-binary combinations, biserial correlations for binary-normal combinations, and specified correlation for normal-normal combinations. If the resulting correlation matrix is not positive definite, a nearest positive matrix will be used.

## Usage

```
compute.sigma.star(no.bin, no.nor, prop.vec.bin = NULL,
 corr.vec = NULL, corr.mat = NULL)
```

## Arguments

| | |
|---|---|
| `no.bin` | Number of binary variables |
| `no.nor` | Number of normal variables |
| `prop.vec.bin` | Probability vector for binary variables |
| `corr.vec` | Vector of elements below the diagonal of correlation matrix ordered columnwise |
| `corr.mat` | Specified correlation matrix |

## Value

| | |
|---|---|
| `sigma_star` | A resulting intermediate correlation matrix $\Sigma^*$ |
| `nonPD` | If a resulting intermediate correlation matrix is non-positive definite, it is stored in this value. Otherwise it is NULL. |
| `PD` | TRUE if $\Sigma^*$ is positive definite, FALSE otherwise. A FALSE indicates that the nearest positive definite matrix is returned. |
| `eigenv` | Eigenvalues of the $\Sigma^*$ before the conversion |

## See Also

`validation.corr`, `nearPD`, `phi2tetra`, `is.positive.definite`,
`jointly.generate.binary.normal`, `simulation`

## Examples

```
cmat = lower.tri.to.corr.mat(corr.vec= c(0.16, 0.04, 0.38, 0.14, 0.47, 0.68),4)
compute.sigma.star(no.bin=2, no.nor=2, prop.vec.bin=c(0.4,0.7),
corr.vec=NULL,corr.mat=cmat)
```

`jointly.generate.binary.normal`
*Generates a mix of binary and normal data*

## Description

Generates multiple binary and normal variables simultaneously given marginal characteristics and association structures.

## Usage

```
jointly.generate.binary.normal(no.rows, no.bin,
no.nor, prop.vec.bin = NULL, mean.vec.nor = NULL, var.nor = NULL,
sigma_star = NULL, corr.vec = NULL, corr.mat = NULL,
continue.with.warning = TRUE)
```

## Arguments

| | |
|---|---|
| `no.rows` | Number of rows. |
| `no.bin` | Number of binary variables |
| `no.nor` | Number of normal variables |
| `prop.vec.bin` | Probability vector for binary variables |
| `mean.vec.nor` | Vector of means for normal variables |
| `var.nor` | Vector of variances for normal variables |
| `sigma_star` | Intermediate correlation matrix |
| `corr.vec` | Vector of elements below the diagonal of correlation matrix ordered columnwise |
| `corr.mat` | Specified correlation matrix |
| `continue.with.warning` | TRUE to proceed with the nearest positive definite $\Sigma^*$. FALSE to terminate program execution if $\Sigma^*$ is not positive definite |

## Value

| | |
|---|---|
| `data` | A matrix of generated data. |

## See Also

`compute.sigma.star`, `validation.corr`, `validation.bin`, `validation.nor`, `nearPD`, `simulation`, `rmvnorm`

### Examples

```
no.rows=100
no.bin=2; no.nor=2
mean.vec.nor=c(3,1); var.nor=c(4,2)
prop.vec.bin=c(0.4,0.7)
corr.vec=c(0.16,0.04,0.38,0.14,0.47,0.68);

cmat = lower.tri.to.corr.mat(corr.vec,4)
sigma.star=compute.sigma.star(no.bin=2, no.nor=2, prop.vec.bin=c(0.4,0.7),
corr.mat=cmat)
mydata=jointly.generate.binary.normal(no.rows,no.bin,no.nor,prop.vec.bin,
mean.vec.nor,var.nor, sigma_star=sigma.star$sigma_star,
continue.with.warning=TRUE)
```

---

## `lower.tri.to.corr.mat` *Converts a vector of correlations to a full correlation matrix*

---

### Description

This function creates full correlation matrix from the vector containing elements below the diagonal.

### Usage

```
lower.tri.to.corr.mat(corr.vec = NULL, d)
```

### Arguments

`corr.vec` A vector of elements below diagonal of correlation matrix. The elements must be ordered starting from first element below diagonal of the first column, then second element below diagonal of the first column and so on.

`d` Number of column in final correlation matrix.

### Value

`corr.mat` Full correlation matrix

### See Also

`lower.tri`

### Examples

```
corr.vec=c(0.16,0.04,0.38,0.14,0.47,0.68)
lower.tri.to.corr.mat(corr.vec,4)
```

simulation *Repeats the data generation process in a simulation scheme*

## Description

Simulates many versions of mixed data, and reports averaged proportion, mean, variance and correlation estimates across replications.

## Usage

```
simulation(seed = NULL, nsim, no.rows, no.bin, no.nor,
mean.vec.nor = NULL, var.nor = NULL, prop.vec.bin = NULL,
corr.vec = NULL, corr.mat = NULL, continue.with.warning = TRUE)
```

## Arguments

| | |
|---|---|
| seed | A seed value for the random number generator. Seed value will be randomly generated unless specified. |
| nsim | Number of simulation runs. |
| no.rows | Number of rows. |
| no.bin | Number of binary variables |
| no.nor | Number of normal variables |
| prop.vec.bin | Probability vector for binary variables |
| mean.vec.nor | Vector of means for normal variables |
| var.nor | Vector of variances for normal variables |
| corr.vec | Vector of elements below the diagonal of correlation matrix ordered columnwise |
| corr.mat | Specified correlation matrix |
| continue.with.warning | TRUE to proceed with the nearest positive definite $\Sigma^*$. FALSE to terminate program execution if $\Sigma^*$ is not positive definite |

## See Also

compute.sigma.star, jointly.generate.binary.normal

## Examples

```
simulation(nsim=10, no.rows=100, no.bin=2, no.nor=2,
mean.vec.nor=c(3,1), var.nor=c(4,2), prop.vec.bin=c(0.4,0.7),
corr.vec=c(0.16,0.04,0.38,0.14,0.47,0.68), corr.mat=NULL)
```

## validation.bin — *Validates the marginal specification of the binary part*

### Description

Checks whether the marginal specification of the binary part is valid and consistent.

### Usage

```
validation.bin(no.bin, prop.vec.bin = NULL)
```

### Arguments

| | |
|---|---|
| `no.bin` | Number of binary variates. |
| `prop.vec.bin` | Probability vector for binary variables |

### Examples

```
## Not run: validation.bin (3,  rep(0.6,4))
validation.bin (4,  rep(0.6,4))
```

## validation.corr — *Validates the specified correlation matrix*

### Description

This function validates the correlation vector and/or matrix for appropriate dimension, symmetry, range, and positive definiteness. If both correlation matrix and correlation vector were supplied, it checks whether the matrix and vector are conformable.

### Usage

```
validation.corr(no.bin, no.nor, prop.vec.bin = NULL,
corr.vec = NULL, corr.mat = NULL)
```

### Arguments

| | |
|---|---|
| `no.bin` | Number of binary variables |
| `no.nor` | Number of normal variables |
| `prop.vec.bin` | Probability vector for binary variables |
| `corr.vec` | Vector of elements below the diagonal of correlation matrix ordered columnwise |
| `corr.mat` | Specified correlation matrix |

### See Also

validation.bin, validation.range

### Examples

```
d=4
corr.vec=c(0.21,0.61,0.78,0.10,0.12,0.65)
corr.mat=lower.tri.to.corr.mat(corr.vec,d)


validation.corr (no.bin=2, no.nor=2,prop.vec.bin=c(0.4,0.7),
corr.vec,corr.mat=corr.mat)
```

---

## validation.nor *Validates the marginal specification of the normal part*

---

### Description

This function checks whether mean and variance parameters for the normal part are valid and consistent.

### Usage

```
validation.nor(no.nor, mean.vec.nor = NULL, var.nor = NULL)
```

### Arguments

| | |
|---|---|
| no.nor | Number of normal variables |
| mean.vec.nor | Vector of means for normal variables |
| var.nor | Vector of variances for normal variables |

---

## validation.range *Checks if the correlation terms are within the feasible range*

---

### Description

This function checks if there are correlation range violations among binary-binary, binary-normal and normal-normal combinations.

### Usage

```
validation.range(no.bin, no.nor, prop.vec.bin = NULL, corr.mat)
```

## Arguments

| | |
|---|---|
| `no.bin` | Number of binary variables |
| `no.nor` | Number of normal variables |
| `prop.vec.bin` | Probability vector for binary variables |
| `corr.mat` | Specified correlation matrix |

## Examples

```
cmat=lower.tri.to.corr.mat(corr.vec=c(0.16,0.04,0.38,0.4,0.47,0.68),4)
validation.range(no.bin=2, no.nor=2, prop.vec.bin=c(0.4,0.7), corr.mat=cmat)
```

# Index