Package ‘mvtBinaryEP’

January 2, 2012

Title Generates Correlated Binary Data

Version 1.0.1

Author Kunthel By and Bahjat Qaqish

Maintainer Kunthel By <kby@bios.unc.edu>

Description Generates correlated binary data based on the method of Emrich and Piedmonte (1991)

Depends mvtnorm

Repository CRAN

Date/Publication 2011-01-25 18:16:02

R topics documented:

`mvtBinaryEP-package` *A Package For Generating Correlated Binary Data*

Description

Uses the algorithm of Emrich and Piedmonte (1991) to generate correlated binary data for a given correlation matrix

Details

Package:	mvtBinaryEP
Type:	Package
Version:	1.0
Date:	2009-02-16
License:	GPL(>=2)

The most important function is `ep`. Other functions include `ranMVN`, `ranMVN2`, and `ranMvnXch`. These are used in the `ep` function to generate correlated binary responses. The user may use these functions if multivariate normal data is desired. We note that the `mvtnorm` package also avails the user of functions to generate multivariate data which may be more efficient than our mutivariate normal functions.

Author(s)

These routines were written by Dr. Bahjat Qaqish in SAS and translated to R by Kunthel By. Questions may be relayed to <kby@bios.unc.edu>

References

Emrich, L.J. and Piedmonte, M.R., A method for generating high-dimensional multivariate binary variates, The American Statistician, 45:302-304, 1991.

`ep` *Correlated Binary Data*

Description

Generates correlated binary data based on the method of Emrich and Piedmonte (1991)

Usage

```
ep(mu, R, rho, n, isd = NULL, nRep = 1, seed = NULL, crit = 1e-06,
maxiter = 20)
```

Arguments

`mu`	Vector of means. If `rho` is specified then `mu` must be of length 1.
`R`	Correlation matrix. If `rho` is specified then `R` is ignored. Only the upper part of R is used.
`rho`	If common mean and exchangeable correlation is desired, then this correlation parameter must be specified.
`n`	Cluster size. If `rho` is specified, then this must be specified as well.

isd	Internal Simulation Descriptor. This is useful for generating more responses based on the parameters used in the prior call to ep. This increases efficiency since the intermediate quantities need not be recomputed. isd is a **list** containing some of the input parameters as well as some intermediate quantities. If this is provided then R and rho are ignored.
nRep	Number of clusters (replications).
seed	Sets the seed
crit	Level of precision used in solving for the tetra-choric correlations.
maxiter	Maximum number of iterations used in solving for the tetra-choric correlations.

Details

The method relies on simulating multivariate normal vectors and then dichotomizing each coordinate. The cutpoints are determined by mu. The correlation matrix S (which are the tetra-choric correlations) of the multivariate normal vectors is computed in such a way that the resulting binary vectors have correlation matrix R. One possible complication is that this process is not always possible. Further, when all tetra-choric correlations are computed, the resulting matrix, S, may not be positive definite. These are two possible failure points in the algorithm; both are detected and reported back by the code.

Value

Returns a list with the following two components:

y	Multivariate response of dimension nRep by length(mu)
isd	Internal Simulation Descriptor

Note

The returned object **isd** is also a list with the following fields:

- mu input parameter
- rho input parameter
- n input parameter
- R input parameter
- rootS The Cholesky root of the tetrachoric correlation matrix S (if positive definite)
- S The tetrachoric correlation matrix S (if NOT positive definite)
- pd Flag, TRUE if S is positive definite, FALSE otherwise
- sp Flag, TRUE if successful in solving for tetrachoric correlations
- i row where solving for the tetrachoric correlation failed, if it did fail
- j column where solving for the tetrachoric correlation failed, if it did fail

See Also

See Also ranMVN, ranMVN2, ranMvnXch

Examples

```
# Create mean vector
mu=c(0.5, 0.3, 0.20, 0.1)

# Create correlation matrix
R = c(
    1    , 0.2 , 0.15, -0.05,
    0.2  , 1   , 0.25, 0.2  ,
    0.15 , 0.25, 1   , 0.25 ,
    -0.05, 0.2 , 0.25, 1
)
R = matrix(R, ncol=4)


ep0 = ep(mu=mu, R=R, nRep=1000, seed=NULL)
apply(ep0$y, 2, mean); cor(ep0$y)

#Generates more responses based on the parameters provided above.
ep1 = ep(isd = ep0$isd, nRep=1000, seed=NULL)
apply(ep1$y, 2, mean); cor(ep1$y)


# 5-variate based on common mean and exchangeable correlation.
ep2 = ep(mu=0.3, rho=0.2, n=5, nRep=10000)
apply(ep2$y, 2, mean); cor(ep2$y)
```

`ranMVN` *Multivariate Normal Data*

Description

Generates multivariate normal data based on a specified covariance matrix.

Usage

```
ranMVN(nRep = 1, Sigma, seed = NULL)
```

Arguments

`nRep`	Numer of clusters (replications)
`Sigma`	Specified covariance matrix
`seed`	Initialize random number generator

Details

The returned matrix of responses, call it Y, has mean 0.

Value

Returns a matrix of response variates of dimension `nRep` by `nrow(Sigma)`

See Also

See Also `ranMVN2`, `ranMvnXch`, `ep`

Examples

```
mu = c(25.3, 28.4, 35.7, 50.2)
c1 = c(17, 11, 8, 10)
c2 = c(11, 17, 9, 8)
c3 = c(8, 9, 17, 9)
c4 = c(10, 8, 9, 17)
S = rbind(c1, c2, c3, c4)


y = mu + ranMVN(nRep = 25, S, seed = NULL)
```

`ranMVN2` *Multivariate Normal Data*

Description

Generates multivariate normal data based on the Cholesky root of the covariance matrix.

Usage

```
ranMVN2(nRep = 1, rootS, seed = NULL)
```

Arguments

`nRep`	Numer of clusters (replications)
`rootS`	Cholesky root of the desired covariance matrix
`seed`	Initialize random number generator

Details

The matrix of response, call it Y, has mean 0.

Value

Returns a matrix of response variates of dimension `nRep` by `nrow(rootS)`

See Also

See Also `ranMVN`, `ranMvnXch`, `ep`

Examples

```
mu = c(25.3, 28.4, 35.7, 50.2)
c1 = c(17, 11, 8, 10)
c2 = c(11, 17, 9, 8)
c3 = c(8, 9, 17, 9)
c4 = c(10, 8, 9, 17)
S = rbind(c1, c2, c3, c4)

rootS=chol(S)

y = mu + ranMVN2(nRep = 25, rootS, seed = NULL)
```

---

ranMvnXch *Multivariate Normal Data*

Description

Generates multivariate normal data based on an EXCHANGEABLE correlation matrix.

Usage

```
ranMvnXch(rho, n, nRep = 1, seed = NULL)
```

Arguments

`rho`	correlation
`n`	cluster size
`nRep`	number of clusters (replications)s
`seed`	initializes the random number generator

Details

This is more efficient than `ranMVN` and `ranMVN2` for exchangeable correlation. The returned matrix of responses, call it Y, has mean 0. We require that `rho` is greater than zero.

Value

Returns a matrix of response variates of dimension `nRep` by `n`

See Also

See Also `ranMVN`, `ranMVN2`, `ep`

Index