

Package ‘bindata’

February 14, 2012

Version 0.9-18

Date 2011-09-18

Title Generation of Artificial Binary Data

Author Friedrich Leisch and Andreas Weingessel and Kurt Hornik

Maintainer Friedrich Leisch <Friedrich.Leisch@R-project.org>

Description Generation of correlated artificial binary data.

License GPL-2

Depends e1071, mvtnorm (>= 0.7-0)

Repository CRAN

Date/Publication 2011-09-18 07:36:43

R topics documented:

bincorr2commonprob	2
check.commonprob	3
commonprob2sigma	4
condprob	5
ra2ba	5
rmvbin	6
simul.commonprob	7
SimulVals	8

Index	10
--------------	-----------

bincorr2commonprob *Convert Binary Correlation Matrix to Matrix of Joint Probabilities*

Description

Compute a matrix of common probabilities for a binary random vector from given marginal probabilities and correlations.

Usage

```
bincorr2commonprob(margprob, bincorr)
```

Arguments

margprob vector of marginal probabilities.
bincorr matrix of binary correlations.

Value

The matrix of common probabilities. This has the probabilities that variable i equals 1 in element (i, i) , and the joint probability that variables i and j both equal 1 in element (i, j) (if $i \neq j$).

Author(s)

Friedrich Leisch

References

Friedrich Leisch, Andreas Weingessel and Kurt Hornik (1998). On the generation of correlated artificial binary data. Working Paper Series, SFB “Adaptive Information Systems and Modelling in Economics and Management Science”, Vienna University of Economics, <http://www.wu-wien.ac.at/am>

See Also

[commonprob2sigma](#), [simul.commonprob](#).

check.commonprob	<i>Check Joint Binary Probabilities</i>
------------------	---

Description

The main diagonal elements `commonprob[i, i]` are interpreted as probabilities p_{A_i} that a binary variable A_i equals 1. The off-diagonal elements `commonprob[i, j]` are the probabilities $p_{A_i A_j}$ that both A_i and A_j are 1.

This program checks some necessary conditions on these probabilities which must be fulfilled in order that a joint distribution of the A_i with the given probabilities can exist.

The conditions checked are

$$0 \leq p_{A_i} \leq 1$$

$$\max(0, p_{A_i} + p_{A_j} - 1) \leq p_{A_i A_j} \leq \min(p_{A_i}, p_{A_j}), i \neq j$$

$$p_{A_i} + p_{A_j} + p_{A_k} - p_{A_i A_j} - p_{A_i A_k} - p_{A_j A_k} \leq 1, i \neq j, i \neq k, j \neq k$$

Usage

```
check.commonprob(commonprob)
```

Arguments

`commonprob` Matrix of pairwise probabilities.

Value

`check.commonprob` returns TRUE, if all conditions are fulfilled. The attribute "message" of the return value contains some information on the errors that were found.

Author(s)

Andreas Weingessel

References

Friedrich Leisch, Andreas Weingessel and Kurt Hornik (1998). On the generation of correlated artificial binary data. Working Paper Series, SFB "Adaptive Information Systems and Modelling in Economics and Management Science", Vienna University of Economics, <http://www.wu-wien.ac.at/am>

See Also

[simul.commonprob](#), [commonprob2sigma](#)

Examples

```
check.commonprob(cbind(c(0.5, 0.4), c(0.4, 0.8)))  
check.commonprob(cbind(c(0.5, 0.25), c(0.25, 0.8)))  
check.commonprob(cbind(c(0.5, 0, 0), c(0, 0.5, 0), c(0, 0, 0.5)))
```

commonprob2sigma	<i>Calculate a Covariance Matrix for the Normal Distribution from a Matrix of Joint Probabilities</i>
------------------	---

Description

Computes a covariance matrix for a normal distribution which corresponds to a binary distribution with marginal probabilities given by `diag(commonprob)` and pairwise probabilities given by `commonprob`.

For the simulations the values of `simulvals` are used.

If a non-valid covariance matrix is the result, the program stops with an error in the case of NA arguments and yields a warning message if the matrix is not positive definite.

Usage

```
commonprob2sigma(commonprob, simulvals)
```

Arguments

<code>commonprob</code>	matrix of pairwise probabilities.
<code>simulvals</code>	array received by <code>simul.commonprob</code> .

Value

A covariance matrix is returned with the same dimensions as `commonprob`.

Author(s)

Friedrich Leisch

References

Friedrich Leisch, Andreas Weingessel and Kurt Hornik (1998). On the generation of correlated artificial binary data. Working Paper Series, SFB “Adaptive Information Systems and Modelling in Economics and Management Science”, Vienna University of Economics, <http://www.wu-wien.ac.at/am>

See Also

[simul.commonprob](#)

Examples

```
m <- cbind(c(1/2,1/5,1/6),c(1/5,1/2,1/6),c(1/6,1/6,1/2))
sigma <- commonprob2sigma(m)
```

condprob

*Conditional Probabilities of Binary Data***Description**

Returns a matrix containing the conditional probabilities $P(x_i = 1|x_j = 1)$ where x_i corresponds to the i -th column of x .

Usage

```
condprob(x)
```

Arguments

x matrix of binary data with rows corresponding to cases and columns corresponding to variables.

Author(s)

Friedrich Leisch

ra2ba

*Convert Real Valued Array to Binary Array***Description**

Converts all values of the real valued array x to binary values by thresholding at 0.

Usage

```
ra2ba(x)
```

Arguments

x array of arbitrary dimension

Author(s)

Friedrich Leisch

Examples

```
x <- array(rnorm(10), dim=c(2,5))
ra2ba(x)
```

`rmvbin`*Multivariate Binary Random Variates*

Description

Creates correlated multivariate binary random variables by thresholding a normal distribution. The correlations of the components can be specified either as common probabilities, correlation matrix of the binary distribution, or covariance matrix of the normal distribution.

Usage

```
rmvbin(n, margprob, commonprob=diag(margprob),  
       bincorr=diag(length(margprob)),  
       sigma=diag(length(margprob)),  
       colnames=NULL, simulvals=NULL)
```

Arguments

<code>n</code>	number of observations.
<code>margprob</code>	margin probabilities that the components are 1.
<code>commonprob</code>	matrix of probabilities that components <i>i</i> and <i>j</i> are simultaneously 1.
<code>bincorr</code>	matrix of binary correlations.
<code>sigma</code>	covariance matrix for the normal distribution.
<code>colnames</code>	vector of column names for the resulting observation matrix.
<code>simulvals</code>	result from <code>simul.commonprob</code> , a default data array is automatically loaded if this argument is omitted.

Details

Only one of the arguments `commonprob`, `bincorr` and `sigma` may be specified. Default are uncorrelated components.

`n` samples from a multivariate normal distribution with mean and variance chosen in order to get the desired margin and common probabilities are sampled. Negative values are converted to 0, positive values to 1.

Author(s)

Friedrich Leisch

References

Friedrich Leisch, Andreas Weingessel and Kurt Hornik (1998). On the generation of correlated artificial binary data. Working Paper Series, SFB “Adaptive Information Systems and Modelling in Economics and Management Science”, Vienna University of Economics, <http://www.wu-wien.ac.at/am>

See Also

[commonprob2sigma](#), [check.commonprob](#), [simul.commonprob](#)

Examples

```
## uncorrelated columns:
rmvbin(10, margprob=c(0.3,0.9))

## correlated columns
m <- cbind(c(1/2,1/5,1/6),c(1/5,1/2,1/6),c(1/6,1/6,1/2))
rmvbin(10,commonprob=m)

## same as the second example, but faster if the same probabilities are
## used repeatedly (commonprob2sigma rather slow)
sigma <- commonprob2sigma(m)
rmvbin(10,margprob=diag(m),sigma=sigma)
```

simul.commonprob

Simulate Joint Binary Probabilities

Description

Compute common probabilities of binary random variates generated by thresholding normal variates at 0.

Usage

```
simul.commonprob(margprob, corr=0, method="integrate", n1=10^5, n2=10)
```

Arguments

margprob	vector of marginal probabilities.
corr	vector of correlation values for normal distribution.
method	either "integrate" or "monte carlo".
n1	number of normal variates if method is "monte carlo".
n2	number of repetitions if method is "monte carlo".

Details

The output of this function is used by [rmvbin](#). For all combinations of `marginprob[i]`, `marginprob[j]` and `corr[k]`, the probability that both components of a normal random variable with mean `qnorm(marginprob[c(i, j)])` and correlation `corr[k]` are larger than zero is computed.

The probabilities are either computed by numerical integration of the multivariate normal density, or by Monte Carlo simulation.

For normal usage of [rmvbin](#) it is not necessary to use this function, one simulation result is provided as variable [SimulVals](#) in this package and loaded by default.

Value

`simul.commonprob` returns an array of dimension $c(\text{length}(\text{margprob}), \text{length}(\text{margprob}), \text{length}(\text{corr}))$.

Author(s)

Friedrich Leisch

References

Friedrich Leisch, Andreas Weingessel and Kurt Hornik (1998). On the generation of correlated artificial binary data. Working Paper Series, SFB “Adaptive Information Systems and Modelling in Economics and Management Science”, Vienna University of Economics, <http://www.wu-wien.ac.at/am>

See Also

[rmvbin](#)

Examples

```
simul.commonprob(seq(0,1,0.5), seq(-1,1,0.5), meth="mo", n1=10^4)

data(SimulVals)
```

SimulVals

Pre-simulated Joint Binary Probabilities

Description

This variable provides a pre-fabricated result from `simul.commonprob` such that it is normally not necessary to use this (time consuming) function, and is used by `rmvbin`.

Usage

```
SimulVals
```

Author(s)

Friedrich Leisch

References

Friedrich Leisch, Andreas Weingessel and Kurt Hornik (1998). On the generation of correlated artificial binary data. Working Paper Series, SFB “Adaptive Information Systems and Modelling in Economics and Management Science”, Vienna University of Economics, <http://www.wu-wien.ac.at/am>

See Also

[simul.commonprob](#), [rmvbin](#)

Index

*Topic **array**

ra2ba, 5

*Topic **distribution**

bincorr2commonprob, 2

check.commonprob, 3

commonprob2sigma, 4

condprob, 5

rmvbin, 6

simul.commonprob, 7

*Topic **multivariate**

bincorr2commonprob, 2

check.commonprob, 3

commonprob2sigma, 4

rmvbin, 6

simul.commonprob, 7

*Topic **sysdata**

SimulVals, 8

bincorr2commonprob, 2

check.commonprob, 3, 7

commonprob2sigma, 2, 3, 4, 7

condprob, 5

ra2ba, 5

rmvbin, 6, 7–9

simul.commonprob, 2–4, 7, 7–9

SimulVals, 7, 8