Package 'MissMech'

April 14, 2015

Type Package

Title Testing Homoscedasticity, Multivariate Normality, and Missing Completely at Random

Version 1.0.2

Date 2015-04-13

Author Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

Maintainer Mortaza Jamshidian <mori@fullerton.edu>

Description To test whether the missing data mechanism, in a set of incompletely observed data, is one of missing completely at random (MCAR).
For detailed description see Jamshidian, M. Jalal, S., and Jansen, C. (2014). ``MissMech: An R Package for Testing Homoscedasticity, Multivariate Normality, and Missing Completely at Random (MCAR)," Journal of Statistical Software, 56(6), 1-31. URL http://www.jstatsoft.org/v56/i06/.

Imports graphics

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2015-04-14 07:30:05

R topics documented:

`MissMech-package` *Testing Homoscedasticity, Multivariate Normality, and Missing Completely at Random*

Description

The main purpose of this package is to test whether the missing data mechanism, in a set of incompletely observed data, is one of missing completely at random (MCAR). As a by-product, however, this package can impute incomplete data, is able to perform a test to determine whether data have a multivariate normal distribution or whether the covariances for several populations are equal. The test of MCAR follows the methodology proposed by Jamshidian and Jalal (2010). It is based on testing equality of covariances between groups consisting of identical missing data patterns. The data are imputed, using two options of normality and distribution free, and the test of equality of covariances between groups with identical missing data patterns is performed also with options of assuming normality (Hawkins test) or non-parametrically. The user, can optionally use her own method of data imputation as well. Multiple imputation is an option as a diagnostic tool to help identify cases or variables that contribute to rejection of MCAR, when the MCAR test is rejecetd (See Jamshidian and Jalal, 2010 for details). As explained in Jamshidian, Jalal, and Jansen (2014), this package can also be used for imputing missing data, test of multivariate normality, and test of equality of covariances between several groups when data are complete.

Details

Package:	MissMech
Type:	Package
Version:	1.0.2
Date:	2015-04-13

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

Maintainer: Mortaza Jamshidian <mori@fullerton.edu>

References

Jamshidian, M. and Jalal, S. (2010). "Tests of homoscedasticity, normality, and missing at random for incomplete multivariate data," *Psychometrika,* 75, 649-674.

Jamshidian, M. Jalal, S., and Jansen, C. (2014). " MissMech: An R Package for Testing Homoscedasticity, Multivariate Normality, and Missing Completely at Random (MCAR)," *Journal of Statistical Software,* 56(6), 1-31.

"URL http://www.jstatsoft.org/v56/i06/."

agingdata *Montpetit and Bergeman Longitudinal Study on Aging Data*

Description

The data consist of 521 cases and 7 variables with 280 of the cases being complete. The variables are Education, Income, Perceived satisfaction of social support, Social coping, Total life events scale, Depression scale, and Self-rated help.

Usage

```
agingdata
```

Format

A data frame with 521 observations on the following 7 variables.

References

Montpetit A, Bergeman CS (2007). "Dimensions of control: Mediational analyses of the stress Chealth relationship." *Personality and Individual Differences,* 43, 2237 - 2248.

AndersonDarling *K-Sample Anderson Darling Test*

Description

This is a non-parametric K-sample test that tests equality of distribution of a variable between k populations based on samples from each of the populations.

Usage

```
AndersonDarling(data, number.cases)
```

Arguments

`data`	A single vector consisting of concatenation of the k samples data being used for the test
`number.cases`	A vector consisting of the number of cases in samples 1, 2, ..., k, respectively

Details

The data is a vector including all the k samples to be used for the test. The j-th element of number.cases is the number of cases in sample j (included in data), for j= 1,...,k.

Value

pn	The test's p-value.
adk.all	The Anderson Darling test statistic corresponding to each group.
adl	The sum of elements of adk.all.
var.sdk	The variance of the finite sample distribution of the Anderson Darling test statistic under the null.

Note

The test does not adjust for tie observations.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Scholz, F.W. and Stephens, M.A. (1987). "K-Sample Anderson-Darling Tests," *Journal of the American Statistical Association*, 82, 918-924.

Examples

```
#---- Example 1
set.seed(50)
n1 <- 30
n2 <- 45
n3 <- 60
v1 <- rnorm(n1)
v2 <- runif(n2)
v3 <- rnorm(n3, 2, 3)
AndersonDarling(data = c(v1, v2, v3), number.cases=c(n1, n2, n3))

#---- Example 2
set.seed(50)
n1 <- 30
n2 <- 45
n3 <- 60
v1 <- rt(n1,4)
v2 <- rt(n2,4)
v3 <- rt(n3,4)
AndersonDarling(data=c(v1, v2, v3), number.cases=c(n1, n2, n3))
```

Ddf *Hessian of the observed datat Multivariate Normal Log-Likelihood with Incomplete Data*

Description

The Hessian of the normal-theory observed data log-likelihood function, evaluated at a given value of the mean vector and the covariance matrix, when data are incomplete. The output is a symmetric matrix with rows/columns corresponding to elements in the mean vector and lower diagonal of the covariance matrix.

Usage

```
Ddf(data, mu, sig)
```

Arguments

data	A matrix consisting of at least two columns. Values must be numerical with missing data indicated by NA.
mu	A row matrix consisting of the values of the mean at which points the Hessian of the log-likelihood is to be computed
sig	A symmetric covariance matrix at at which points the Hessian of the log-likelihood is to be computed

Details

While mu is a vector, it has to be input as a matrix object. See example nelow.

Value

dd	The resulting Hessian matrix
se	Negative of the inverse of the Hessian matrix

Note

There must be no rows in data that contain no observations.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Jamshidian, M. and Bentler, P. M. (1999). "ML estimation of mean and covariance structures with missing data using complete data routines." *Journal of Educational and Behavioral Statistics,* 24, 21-41.

Examples

```
set.seed <- 50
n <- 200
p <- 4
pctmiss <- 0.2
y <- matrix(rnorm(n * p),nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA
mu <- c(0,0,0,0)
sig <- matrix(c(1,0,0,0, 0,1,0,0, 0,0,1,0, 0,0,0,1),4,4)
Ddf(data=y, as.matrix(mu), sig)
```

`DelLessData` *Removes groups with identical missing data patterns having at most a given number of cases*

Description

Removes groups of missing data patterns with number of cases less than or equal to a specified value (ncases).

Usage

```
DelLessData(data, ncases = 0)
```

Arguments

`data`	A matrix consisting of at least two columns. Values must be numerical with missing data indicated by NA.
`ncases`	Missing data pattern groups with ncases number of cases or less will be removed from the data set

Value

`data`	A matrix of reaaranged data, according to missing data patterns, with missing data patterns having less than ncases number of cases removed.
`patused`	A matrix indicating the missing data patterns in the data set; observed variable(s) are indiated by 1's' and missing variables are indicated by NA's.
`patcnt`	A vector consisting of the number of cases corresponding to each pattern in patused.
`spatcnt`	Cumulative sum of elements of patcnt.
`g`	Number of missing data patterns.
`caseorder`	A mapping of case number indices from output data (rearranged data) to input data.
`removedcases`	The index of cases that were removed from the original data set

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

Examples

```
set.seed <- 50
n <- 200
p <- 4
pctmiss <- 0.2
y <- matrix(rnorm(n * p),nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA
out <- DelLessData(data=y, ncases = 4)
dim(y)
dim(out$data)
```

Hawkins *Test Statistic for the Hawkins Homoscedasticity Test*

Description

Produces the F_ij's and A_ij's that are used in the Hawkins' test of homogeneoity of covariances. See Hawkins (1981) and Jamshidian and Jalal (2010) for more details.

Usage

```
Hawkins(data, spatcnt)
```

Arguments

data	A matrix consisting of at least two columns and two rows.
spatcnt	The cumulative sum of the number of cases corresponding to each group.

Value

fij	A vector of F_ij statistics, see Hawkins (1981) reference.
a	A list containg A_ij statistics, see Hawkins (1981) reference.
ni	A vector consisting of the number of cases in each group.

Note

There must be no rows in data that contain no observations.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Hawkins, D. M. (1981). "A new test for multivariate normality and homoscedasticity," *Technometrics,* 23, 105-110.

Jamshidian, M. and Jalal, S. (2010). "Tests of homoscedasticity, normality, and missing at random for incomplete multivariate data," *Psychometrika,* 75, 649-674.

Examples

```
set.seed <- 50
n <- 200
p <- 4
pctmiss <- 0.2
y <- matrix(rnorm(n * p),nrow = n)
spatcnt <- c(20, 50, 70, 200)
h <- Hawkins(data=y, spatcnt)
```

Impute *Parametric and Non-Parameric Imputation*

Description

This function imputes the data using two methods.

method 'Normal' - Imputes the data assuming that the data come from a multivariate normal distribution with mean mu and covariance sig. If mu or sig are not inputted, then their maximum likelihood estimate is used. The imputed values are based on the conditional distribution of the missing given the observed and mu and sigma; see Jamshidian and Jalal (2010) for more details.

method 'Dist.Free' - This method imputes the data nonparametrically using the method of Sirvastava and Dolatabadi (2009). Also see Jamshidian and Jalal (2010).

Usage

```
Impute(data, mu = NA, sig = NA, imputation.method = "Normal", resid = NA)
```

Arguments

`data`	A matrix consisting of at least two columns. Values must be numerical with missing data indicated by NA.
`mu`	A vector, consisting of population means, used to impute the data. As a default the maximum likelihood estimates based on the observed data is used.
`sig`	The population covariance matrix used to impute the data. As a default the maximum likelihood estimates based on the observed data is used.
`imputation.method`	'normal' uses the normal imputation method. 'dist.free uses the the method. See Jamshidian and Jalal (2010) and Sirvastava and Dolatabadi (2009).
`resid`	User defined residual vector to be used in place of the residuals proposed by the Sirvastava and Dolatabadi (2009) method.

Details

This routine uses OrderMissing to order data accordinng to missing data patterns. The output consists of imputed data both in its original order as well as post ordering by OrderMissing.

Value

`yimp`	The imputed data set (in the order of the original data) after rwos with no datum (if any) have been deleted.
`yimpOrdered`	The imputed data set ordered by OrderMissing according to missing data pattern
`caseorder`	A mapping of case number indices from OrderedData to the original data. More specifically, the j-th row of the OrderedData is the caseorder[j]-th (the j-th element of caseorder) row of the original data.
`patused`	A matrix indicating the missing data patterns in the data set, using 1's' (for observed) and NA's (for missing).
`patcnt`	A vector consisting the number of cases corresponding to each pattern in patused.

Note

In the above descriptions "original data" refers to the input data after deletion of the rows consisting of all NA's (if any)

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Srivastava, M. S. and Dolatabadi, M. (2009). "Multiple imputation and other resampling scheme for imputing missing observations," Journal of Multivariate Analysis, 100, 1919-1937.

Jamshidian, M. and Jalal, S. (2010). "Tests of homoscedasticity, normality, and missing at random for incomplete multivariate data," *Psychometrika,* 75, 649-674.

Examples

```
set.seed <- 50
n <- 200
p <- 4
pctmiss <- 0.2
y <- matrix(rnorm(n * p),nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA

yimp1 <- Impute(data=y, mu = NA, sig = NA, imputation.method = "Normal", resid = NA)
yimp2 <- Impute(data=y, mu = NA, sig = NA, imputation.method = "Dist.Free", resid = NA)
```

LegNorm *Evaluating Legendre's Polynomials of Degree 1, 2, 3, or 4*

Description

This function evaluates the values of Legendre polynomials of degrees 1 to 4 on [0,1] at a value(s) x.

Usage

```
LegNorm(x)
```

Arguments

x	A scaler or vector of values at which the Legendre's ploynomials are to be evaluated.

Value

The returned list has the following elements:

p1	p1 is value(s) of the Legendre's polynomial of degree 1 at x
p2	p2 is value(s) of the Legendre's polynomial of degree 2 at x
p3	p3 is value(s) of the Legendre's polynomial of degree 3 at x
p4	p4 is value(s) of the Legendre's polynomial of degree 4 at x

Note

Legendre's polynomias on [0,1] are calculated.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

David, F. N. (1939). "On Neyman's "smooth" test for goodness of fit: I. Distribution of the criterion Psi2 when the hypothesis tested is true," *Biometrika,* 31, 191-199.

Examples

```
p <- LegNorm(c(5.2,11,15))
p$p3
```

`Mls` *ML Estimates of Mean and Covariance Based on Incomplete Data*

Description

Normal theory maximum likelihood estimates of mean and covariance matrix is obtained when data are incomplete, using EM algorithm (see Jamshidian and Bentler 1999). If the option Hessian is set to TRUE, then the observed information containing the standard errors of the parameter estimates is also computed.

Usage

```
Mls(data, mu = NA, sig = NA, tol = 1e-06, Hessian = FALSE)
```

Arguments

`data`	A matrix consisting of at least two columns. Values must be numerical with missing data indicated by NA.
`mu`	EM iteration initial value for mu (the mean vector). The default is a zero vector.
`sig`	EM iteration initial value for sigma (the covariance matrix). The default is the identity matrix.
`tol`	The tolerance value used in the convergence criteria described in Jamshidian and Bentler (1999) for stopping the EM algorithm.
`Hessian`	Hessian of the log-likelihood function, see Jamshidian and Bentler (1999).

Value

`mu`	The maximum likelihood estimate of the mean vector.
`sig`	The maximum likelihood estimates of the covariance matrix.
`hessian`	The Hessian of the observed data log-likelihood function.
`stderror`	The negative of the inverse of the Hessian of the observed data log-likelihood function. The diagonal elements of stderror are the variance of the parameters based on the observed informaion matrix.
`iteration`	The number of iterations used in the EM iterative process.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Jamshidian, M. and Bentler, P. M. (1999). "ML estimation of mean and covariance structures with missing data using complete data routines." *Journal of Educational and Behavioral Statistics,* 24, 21-41.

Examples

```
set.seed <- 50
n <- 200
p <- 4
pctmiss <- 0.2 # Generate 20% missing data
y <- matrix(rnorm(n * p),nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA
ml <- Mls(data=y, mu = NA, sig = NA, tol = 1e-6, Hessian=FALSE)
ml
```

`OrderMissing` *Order Missing Data Pattern*

Description

This function rearranges the data based on their missing data patterns. Morever, missing data patterns consisting of fewer than the user specified number in del.lesscases is deleted from the dataset.

Usage

```
OrderMissing(data, del.lesscases = 0)
```

Arguments

`data`	A matrix or data frame consisting of at least two columns. Values must be numerical with missing data indicated by NA.
`del.lesscases`	Missing data patterns consisting of del.lesscases number of cases or less will be removed from the data set.

Value

An object of class `orderpattern` - a list including elements:

`data`	A matrix containing the rearranged data set
`caseorder`	A mapping of case number indices from output (ordered) data to input data. More specifically, the j-th row of the output (ordered) data is the caseorder[j]-th (the j-th element of caseorder) row of the input data.
`patused`	A matrix indicating the missing data patterns in the data set, using 1's' (for observed) and NA's (for missing)
`patcnt`	A vector consisting the number of cases corresponding to each pattern in patused
`spatcnt`	Cumulative sum of elements of patcnt
`g`	Number of missing data patterns
`removedcases`	The index of cases that were removed from the original data set

Note

If you run the following command line: out <- OrderMissing(data=y, del.lesscases=0), then y[out$caseorder,] is equal to the output data (i.e. out$data). Also out$data[order(out$caseorder),] is equal to the original data y. Note that if del.lesscases is greater than zero, the command out$data[order(out$caseorder),] will result in a data set that has no case order correspondnce to the original data.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

Examples

```
set.seed <- 50
n <- 200
p <- 4
pctmiss <- 0.2
y <- matrix(rnorm(n * p),nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA
out <- OrderMissing(y, del.lesscases = 0)
a <- out$caseorder
z = out$data
y[a,] # Reverting the original data to the new output order
z[order(a),] # Reverting the ordered datat to the original order
```

`TestMCARNormality` *Testing Homoscedasticity, Multivariate Normality, and Missing Completely at Random*

Description

The main purpose of this package is to test whether the missing data mechanism, for an incompletely observed data set, is one of missing completely at random (MCAR). As a by product, however, this package has the capabilities of imputing incomplete data, performing a test to determine whether data have a multivariate normal distribution, performing a test of equality of covariances for groups, and obtaining normal-theory maximum likelihood estimates for mean and covariance when data are incomplete. The test of MCAR follows the methodology proposed by Jamshidian and Jalal (2010). It is based on testing equality of covariances between groups having identical missing data patterns. The data are imputed, using two options of normality and distribution free, and the test of equality of covariances between groups with identical missing data patterns is performed also with options of assuming normality (Hawkins test) or non-parametrically. Users can optionally use their own method of data imputation as well. Multiple imputation is an additional feature of the program that can be used as a diagnostic tool to help identify cases or variables that contribute to rejection of MCAR, when the MCAR test is rejecetd (See Jamshidian and Jalal, 2010 for details). As explained in Jamshidian, Jalal, and Jansen (2014), this package can also be used for imputing missing data, test of multivariate normality, and test of equality of covariances between several groups when data are completly observed.

Usage

```
TestMCARNormality(data, del.lesscases = 6, imputation.number = 1, method =
                  "Auto", imputation.method = "Dist.Free", nrep = 10000,
                  n.min = 30, seed = 110, alpha = 0.05, imputed.data = NA)
```

Arguments

data	A matrix or data frame consisting of at least two columns. Values must be numerical with missing data indicated by NA.
del.lesscases	Missing data patterns consisting of del.lesscases number of cases or less will be removed from the data set.
imputation.number	Number of imputations to be used, if data are to be multiply imputed.
method	method is an option that allows the user to select one of the methods of Hawkins or nonparametric for the test. If the user is certain that data have multivariate normal distribution, the method="Hawkins" should be selected. On the other hand if data are not normally distributed, then method="Nonparametric" should be used. If the user is unsure, then the default value of method="Auto" will be used, in which case both the Hawkins and the nonparametric tests will be run, and the default output follows the recommendation by Jamshidian and Jalal (2010) outlined in their flowchart given in Figure 7 of their paper.
imputation.method	"Dist.Free": Missing data are imputed nonparametrically using the method of Sirvastava and Dolatabadi (2009); also see Jamshidian and Jalal (2010). "normal": Missing data are imputed assuming that the data come from a multivariate normal distribution. The maximum likelihood estimate of the mean and covariance obtained from Mls is used for generating imputed values. The imputed values are based on the conditional distribution of the missing variables given the observed variables; see Jamshidian and Jalal (2010) for more details.
nrep	Number of replications used to simulate the Neyman distribution to determine the cut off value for the Neyman test in the program SimNey. Larger values increase the accuracy of the Neyman test.
n.min	The minimum number of cases in a group that triggers the use of asymptotic Chi distribution in place of the emprical distribution in the Neyman test of uniformity.
seed	An initial random number generator seed. The default is 110 that can be reset to a user selected number. If the value is set to NA, a system selected seed is used.
alpha	The significance level at which tests are performed.
imputed.data	The user can optionally provide an imputed data set. In this case the program will not impute the data and will use the imputed data set for the tests performed. Note that the order of cases in the imputed data set should be the same as that of the incomplete data set.

Details

Theoretical, technical and prcatical details about this program and its uses can be found in Jamshidian and Jalal (2010) and Jamshidian, Jalal, and Jansen (2014).

Value

analyzed.data	The data that were used in the analysis. If del.lesscases=0, this is the same as the orginal data inputted. If del.lesscases > 0, then this is the data with cases removed.
imputed.data	The analyzed.data after imputation. If imputation.number > 1, the first imputed data set is returned.
ordered.data	The analyzed.data ordered according to missing data pattern, usin the function OrderMissing.
caseorder	A mapping of case number indices from ordered.data to the original data. More specifically, the j-th row of the ordered.data is the caseorder[j]-th (the j-th element of caseorder) row of the original data.
pnormality	p-value for the nonparametric test: When imputation.number > 1, this is a vector with each element corresponding to each of the imputed data sets.
adistar	A matrix consisting of the Anderson-Darling test statistic for each group (columns) and each imputation (rows).
adstar	Sum of adistar: When imputation.number >1, this is a vector with each element corresponding to each of the imputed data sets.
pvalcomb	p-value for the Hawkins test: When imputation.number >1, this is a vector with each element corresponding to each of the imputed data sets.
pvalsn	A matrix consisting of Hawkins test statistics for each group (columns) and each imputation (rows).
g	Number of patterns used in the analysis.
combp	Hawkins test statistic: When imputation.number > 1, this is a vector with each element corresponding to each of the imputed data sets.
alpha	The significance level at which the hypothesis tests are performed.
patcnt	A vector consisting the number of cases corresponding to each pattern in patused.
patused	A matrix indicating the missing data patterns in the data set, using 1 and NA's.
imputation.number	A value greater than or equal to 1. If a value larger than 1 is used, data will be imputed imputation.number times.
mu	The normal-theory maximum likelihood estimate of the variables means.
sigma	The normal-theory maximum likelihood estimate of the variables covariance matrix.

Note

Note 1: In the above descriptions "original data" refers to the input data after deletion of the rows consisting of all NA's (if any)

Note 2: The normal theory maximum likelihood estimate of mean and covariance is obtained using the EM algorithm, as described in Jamshidian and Bentler (1999). The standard errors for these estimates, based on the observed information matrix, can be obtained via the function Ddf, included in this package.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Jamshidian, M. and Bentler, P. M. (1999). "ML estimation of mean and covariance structures with missing data using complete data routines." *Journal of Educational and Behavioral Statistics,* 24, 21-41.

Jamshidian, M. and Jalal, S. (2010). "Tests of homoscedasticity, normality, and missing at random for incomplete multivariate data," *Psychometrika,* 75, 649-674.

Jamshidian, M. Jalal, S., and Jansen, C. (2014). " MissMech: An R Package for Testing Homoscedasticity, Multivariate Normality, and Missing Completely at Random (MCAR)," *Journal of Statistical Software,* 56(6), 1-31.

Examples

```
#-- Example 1: Data are MCAR and normally distributed
n <- 300
p <- 5
pctmiss <- 0.2
set.seed(1010)
y <- matrix(rnorm(n * p),nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA
out <- TestMCARNormality(data=y)
print(out)

# --- Prints the p-value for both the Hawkins and the nonparametric test
summary(out)

# --- Uses more cases
#out1 <- TestMCARNormality(data=y, del.lesscases = 1)
#print(out1)

#---- performs multiple imputation
Out <- TestMCARNormality (data = y, imputation.number = 10)
summary(Out)
boxplot(Out)


#-- Example 2: Data are MCAR and non-normally distributed (t distributed with d.f. = 5)
n <- 300
p <- 5
pctmiss <- 0.2
set.seed(1010)
y <- matrix(rt(n * p, 5), nrow = n)
missing <- matrix(runif(n * p), nrow = n) < pctmiss
y[missing] <- NA
out <- TestMCARNormality(data=y)
print(out)
```

```
# Perform multiple imputation
#Out_m <- TestMCARNormality (data = y, imputation.number = 20)
#boxplot(Out_m)

# One may impute the data using a method other than the methods available in the package
# MissMech. If object "yimputed" set to be imputed data using other methods, e.g. k nearest
# neighbor imputation, then in MissMech it can be implemented as follow
#out_k <- TestMCARNormality(data = y, imputed.data = yimputed)
#print(out_k)

#-- Example 3: Data are MAR (not MCAR), but are normally distributed
n <- 300
p <- 5
r <- 0.3
mu <- rep(0, p)
sigma <- r * (matrix(1, p, p) - diag(1, p))+ diag(1, p)
set.seed(110)
eig <- eigen(sigma)
sig.sqrt <- eig$vectors %*%  diag(sqrt(eig$values)) %*%  solve(eig$vectors)
sig.sqrt <- (sig.sqrt + sig.sqrt) / 2
y <- matrix(rnorm(n * p), nrow = n) %*%  sig.sqrt
tmp <- y
for (j in 2:p){
  y[tmp[, j - 1] > 0.8, j] <- NA
}
out <- TestMCARNormality(data = y, alpha =0.1)
print(out)

#-- Example 4: Multiple imputation; data are MAR (not MCAR), but are normally distributed
#n <- 300
#p <- 5
#pctmiss <- 0.2
#set.seed(1010)
#y <- matrix (rnorm(n * p), nrow = n)
#missing <- matrix(runif(n * p), nrow = n) < pctmiss
#y[missing] <- NA
#Out <- OrderMissing(y)
#y <- Out$data
#spatcnt <- Out$spatcnt
#g2 <- seq(spatcnt[1] + 1, spatcnt[2])
#g4 <- seq(spatcnt[3] + 1, spatcnt[4])
#y[c(g2, g4), ] <- 2 * y[c(g2, g4), ]
#out <- TestMCARNormality(data = y, imputation.number = 20)
#print(out)
#boxplot(out)

# Removing Groups 2 and 4

#y1= y[-seq(spatcnt[1]+1,spatcnt[2]),]
#out <- TestMCARNormality(data=y1,imputation.number = 20)
#print(out)
#boxplot(out)
```

```
#-- Example 5: Test of homoscedasticity for complete data
#n <- 50
#p <- 5
#r <- 0.4
#sigma <- r * (matrix(1, p, p) - diag(1, p)) + diag(1, p)
#set.seed(1010)
#eig <- eigen(sigma)
#sig.sqrt <- eig$vectors %*%  diag(sqrt(eig$values)) %*%  solve(eig$vectors)
#sig.sqrt <- (sig.sqrt + sig.sqrt) / 2
#y1 <- matrix(rnorm(n * p), nrow = n) %*%  sig.sqrt
#n <- 75
#p <- 5
#y2 <- matrix(rnorm(n * p), nrow = n)
#n <- 25
#p <- 5
#r <- 0
#sigma <- r * (matrix(1, p, p) - diag(1, p)) + diag(2, p)
#y3 <- matrix(rnorm(n * p), nrow = n) %*%  sqrt(sigma)
#ycomplete <- rbind(y1 ,y2 ,y3)
#y1 [ ,1] <- NA
#y2[,c(1 ,3)] <- NA
#y3 [ ,2] <- NA
#ygroup <- rbind(y1, y2, y3)
#out <- TestMCARNormality(data = ygroup, method = "Hawkins", imputed.data = ycomplete)
#print(out)

# ---- Example 6, real data
#data(agingdata)
#TestMCARNormality(agingdata, del.lesscases = 1)
```

`TestUNey` *Test of Goodness of Fit (Uniformity)*

Description

This routine tests whether the values in a vector x is distributed as uniform (0,1). The Neyman's smooth test of fit, as described by Ladwina (1994) is used. The p-values are obtained based on a resampling method from uniform (0,1).

Usage

```
TestUNey(x, nrep = 10000, sim = NA, n.min = 30)
```

Arguments

`x`	A vector of values, each in the interval [0,1].
`nrep`	The number of replications used to simulate the Neyman distribution.

sim	A vector of simulated values from the Neyman distribution. If sim = NA this vector is generated by the function SimNev, otherwise the vector inputted is used.
n.min	The minimum sample size that triggers the use of asymptotic Chi distribution in place of the emprical distribution in the Neyman test of uniformity.

Value

pn	The p-value for the test.
n4	The value of the test statistics.

Author(s)

Mortaza Jamshidian, Siavash Jalal, and Camden Jansen

References

Ledwina, T. (1994). "Data-driven version of neyman's smooth test of fit," *Journal of the American Statistical Association,* 89, 1000-1005.

Examples

```
# Example 1
x <- runif(100)
TestUNey(x, nrep = 10000, sim = NA)

# Example 2
x <- runif(30,2,5)
x <- (x-min(x))/(max(x)-min(x))
TestUNey(x, nrep = 10000, sim = NA)

# Example 3
x <- c(0.6,0.6,0.5,0.7,0.3,0.4,0.5,0.4,0.2,0.4,0.2,0.5,0.7,0.1,0.7,0.1,0.5,0.5,0.4,0.6,0.3)
TestUNey(x, nrep = 10000, sim = NA)
```

Index