Package: multiUS (via r-universe)

October 9, 2024

Type Package

Title Functions for the Courses Multivariate Analysis and Computer Intensive Methods

Version 1.2.3

Description Provides utility functions for multivariate analysis (factor analysis, discriminant analysis, and others). The package is primary written for the course Multivariate analysis and for the course Computer intensive methods at the masters program of Applied Statistics at University of Ljubljana.

License GPL (>= 2)

Encoding UTF-8

Imports stats, gplots, graphics, MASS

Suggests CCA, psych

RoxygenNote 7.2.0

NeedsCompilation no

Author Žiberna Aleš [aut], Cugmas Marjan [cre, aut], Torgo Luis [cph], Ki-Yeol Kim [cph], Gwan-Su Yi [cph], Liaw Andy [cph], Leisch Friedrich [cph]

Maintainer Cugmas Marjan <marjan.cugmas@fdv.uni-lj.si>

Date/Publication 2023-01-23 11:10:02 UTC

Repository https://mrmarjan.r-universe.dev

RemoteUrl https://github.com/cran/multiUS

RemoteRef HEAD

RemoteSha ad6a96a308397a73ed22b8bf44c08ef28380d6a1

Contents

`antiImage` *Anti-image matrix*

Description

The function computes anti-image matrix (i.e., with partial correlations on the off-diagonal and with KMO-MSAs on the diagonal) and the overall KMO.

Usage

```
antiImage(X)
```

Arguments

`X`	A data frame with the values of numerical variables.

Value

A list with two elements:

- `AIR` - Anti-image matrix.
- `KMO` - Overall KMO.

Author(s)

Marjan Cugmas

References

Kaiser, H. F., & Rice, J. (1974). Little Jiffy, Mark Iv. Educational & Psychological Measurement, 34(1), 111.

Examples

```
antiImage(X = mtcars[, c(1, 3, 4, 5)])
```

BoxMTest *Box's test for equivalence of covariance matrices*

Description

The function performs Box's test for testing the null hypothesis that two or more covariance matrices are equal.

Usage

```
BoxMTest(X, cl, alpha = 0.05, test = "any")
```

Arguments

`X`	A data frame with the values of numberical variables.
`cl`	An normial or ordinal variable which defines groups (a partition) (must be of type `factor`).
`alpha`	Significance level (default `0.05`).
`test`	Wheter the F-test (`test = "F"`) or Chi-square (`test = "ChiSq"`) test should be forced (see Details). In the case of default value `any`, the test is chosen based on the number of units by groups.

Details

If the size of any group is at least 20 units (sufficiently large), the test takes a Chi-square approximation, otherwise it takes an F approximation.

Value

A list with the following elements:

- `MBox` - The value of the Box's M statistic.
- `ChiSq` or `F` - The approximation statistic test.
- `p` - An observed significance level.

Author(s)

Andy Liaw and Aleš Žiberna (minor modifications)

References

Stevens, J. (1996). Applied multivariate statistics for the social sciences . 1992. Hillsdale, NJ: Laurence Erlbaum.

Examples

```
BoxMTest(X = mtcars[, c(1, 3, 4, 5)], cl = as.factor(mtcars[, 2]), alpha = 0.05)
```

---

breakString *Break a string*

Description

The function breaks a string after around the specified number of characters.

Usage

```
breakString(x, nChar = 20)
```

Arguments

x	A string.
nChar	The number of characters after which the new line is inserted. Default to 20.

Value

A string with inserted \n.

Author(s)

Marjan Cugmas

Examples

```
someText <- "This is the function that breaks a string."
breakString(x = someText, nChar = 20)
```

cancorPlus *Canonical correlations*

Description

The function computes canonical correlations (by using cc or cancor functions) and provides with the test of canonical correlations and with the eigenvalues of the canonical roots (including with the proportion of explained variances by correlation and other related statistics).

Usage

```
cancorPlus(x, y, xcenter = TRUE, ycenter = TRUE, useCCApackage = FALSE)
```

Arguments

x	A data frame or a matrix with the values that correspond to the first set of variables (X-variables).
y	A data frame or a matrix with the values that correspond to the second set of variables (Y-variables).
xcenter	Whether any centring have to be done on the x values before the analysis. If TRUE (default), subtract the column means. If FALSE, do not adjust the columns. Otherwise, a vector of values to be subtracted from the columns.
ycenter	Analogous to xcenter, but for the y values.
useCCApackage	Whether cc function (from CCA package) or cancor function (from stats package) should be used to obtain canonical correlations.

Value

The function returns the same output as functions cancor or cc with the following additional elements:

- $sigTest
  - WilksL - Value of the Wilk's lambda statistic (it is a generalization of the multivariate R2; values near 0 indicate high correlation while values near 1 indicate low correlation).
  - F - Corresponding (to Wilk's lambda) F-ratio.
  - df1 - Degrees of freedom for the corresponding F-ratio.
  - df2 - Degrees of freedom for the corresponding F-ratio.
  - p - Probability value (p-value) for the corresponding F-ratio (Ho: The current and all the later canonical correlations equal to zero).
- $eigModel
  - Eigenvalues - Eigenvalues of the canonical roots.
  - % - Proportion of explained variance of correlation.
  - Cum % - Cumulative proportion of explained variance of correlation.
  - Cor - Canonical correlation coefficient.
  - Sq. Cor - Squared canonical correlation coefficient.

Author(s)

Adapted by Aleš Žiberna based on the source in References.

References

R Data Analysis Examples: Canonical Correlation Analysis, UCLA: Statistical Consulting Group. From http://www.ats.ucla.edu/stat/r/dae/canonical.htm (accessed Decembar 27, 2013).

See Also

`testCC`

Examples

```
cancorPlus(x = mtcars[, c(1,2,3)], y = mtcars[, c(4,5, 6)])
```

---

compLoad *Compare factor loadings*

Description

The function compares two sets of factor loadings by considering different possible orders of factors and different possible signs of factor loadings.

Usage

```
compLoad(L1, L2)
```

Arguments

`L1`	First set of factor loadings in a matrix form (variables are organized in rows and factors are organized in columns).
`L2`	Second set of factor loadings in a matrix form (variables are organized in rows and factors are organized in columns).

Value

A list with the following elements:

- `err` - Sum of squared differences between the values of `L1` and `L2` (for the corresponding permuation and signs).
- `perm` - Permutation of columns of `L1` that results in the lowest `err` value.
- `sign` - Signs of factor loadings of `L1`. The first value corresponds to the first column of `L1` and the second value corresponds to the second column of `L1`.

Author(s)

Aleš Žiberna and Friedrich Leisch (permutations)

Examples

```
L1 <- cbind(c(0.72, 0.81, 0.92, 0.31, 0.22, 0.15), c(0.11, 0.09, 0.17, 0.77, 0.66, 0.89))
L2 <- cbind(c(-0.13, -0.08, -0.20, -0.78, -0.69, -0.88), c(0.72, 0.82, 0.90, 0.29, 0.20, 0.17))
compLoad(L1, L2)
```

`corTestDf` *Compute correlations and test their statistical significance*

Description

The function computes the whole correlation matrix and corresponding sample sizes and p-values. Print method is also available.

Usage

```
corTestDf(X, method = "p", use = "everything", ...)

## S3 method for class 'corTestDf'
print(x, digits = c(3, 3), format = NULL, ...)

printCorTestDf(l, digits = c(3, 3), format = NULL, ...)
```

Arguments

`X`	Data matrix with selected variables.
`method`	A type of correlation coefficient to be calculated, see function `cor`.
`use`	In the case of missing values, which method should be used, see function `cor`.
`...`	Other parameters to print.default (not needed).
`x`	Output of `corTestDf` function.
`digits`	Vector of length two for the number of digits (the first element of a vector corresponds to the number of digits for correlation coefficients and the second element of a vector corresponds to the number of digits for p-values).
`format`	A vector of length two for the formatting of the output values.
`l`	Output of `corTestDf` function.

Author(s)

Ales Ziberna

See Also

`cor.test`

Examples

```
corTestDf(mtcars[, 3:5])
```

`discretize` *Transform continuous variable to a discrete variable*

Description

The function transforms a continuous variable to a k-point discrete variable (similar to a Likert-item type variable). Different styles of answering to a survey are possible.

Usage

```
discretize(x, type = "eq", q = 1.5, k = 5, r = range(x), num = TRUE)
```

Arguments

`x`	Vector with values to be transformed.
`type`	Type of transformation. Possible values are: `eq` (default) (equal wide intervals), `yes` (wider intervals at higher values of `x`), `no` (wider intervals at lower values of `x`), `avg` (wider intervals near the mean of `x`).
`q`	Extension factor. Tells how much is each next interval wider then the previous one. Not used when `type="eq"`.
`k`	Number of classes.
`r`	Minimum and maximum values to define intervals of `x`. Default are minimum and maximum values of `x`.
`num`	If `TRUE` (default) numerical values are returned, otherwise intervals are returned.

Value

Transformed values are organized into a vector.

Author(s)

Aleš Žiberna

Examples

```
x <- rnorm(1000)
hist(x = discretize(x, type = "eq"), breaks = 0:5+0.5, xlab = "answer", main = "type = 'eq'")
hist(x = discretize(x, type = "yes"), breaks = 0:5+0.5, xlab = "answer", main = "type = 'yes'")
hist(x = discretize(x, type = "no"), breaks = 0:5+0.5, xlab = "answer", main = "type = 'no'")
hist(x = discretize(x, type = "avg"), breaks = 0:5+0.5, xlab = "answer", main = "type = 'avg'")
```

freqTab *Create a frequency table*

Description

The function creates a frequency table with percentages for the selected categorical variable.

Usage

```
freqTab(x, dec = 2, cum = TRUE, ...)
```

Arguments

x	Vector with the values of a categorical variable.
dec	Number of decimal places for percentages.
cum	whether to calculate cumulative frequencies and percentages (default TRUE).
...	Arguments passed to function table.

Value

A frequency table (as a dataframe).

Author(s)

Aleš Žiberna

Examples

```
freqTab(mtcars[,2], dec = 1)
```

histNorm *Histogram with normal curve*

Description

The function draws a histogram with a normal density curve. The parameters (mean and standard deviation) are estimated on the empirical data.

Usage

```
histNorm(y, breaks = "Sturges", freq = TRUE, ...)
```

Arguments

y	A vector of observations.
breaks	See help file for function hist.
freq	Wheter frequencies (freq = TRUE) of density (freq = FALSE) should be represented on y-axis.
...	Arguments passed to function hist.

Value

A list with two elements:

- x - breaks, see graphics::hist.
- y - frequencies or relative frequencies, see graphics::hist.

Author(s)

Marjan Cugmas

Examples

```
histNorm(rnorm(1000), freq = TRUE)
histNorm(rnorm(1000), freq = FALSE)
```

KNNimp *KNN-imputation method*

Description

Function that fills in all NA values using the k-nearest-neighbours of each case with NA values. By default it uses the values of the neighbours and obtains an weighted (by the distance to the case) average of their values to fill in the unknows. If meth='median' it uses the median/most frequent value, instead.

Usage

```
KNNimp(data, k = 10, scale = TRUE, meth = "weighAvg", distData = NULL)
```

Arguments

data	A data frame with the data set.
k	The number of nearest neighbours to use (defaults to 10).
scale	Boolean setting if the data should be scale before finding the nearest neighbours (defaults to TRUE).
meth	String indicating the method used to calculate the value to fill in each NA. Available values are median or weighAvg (the default).

`distData` Optionally you may sepecify here a data frame containing the data set that should be used to find the neighbours. This is usefull when filling in NA values on a test set, where you should use only information from the training set. This defaults to `NULL`, which means that the neighbours will be searched in data.

Details

This function uses the k-nearest neighbours to fill in the unknown (NA) values in a data set. For each case with any NA value it will search for its k most similar cases and use the values of these cases to fill in the unknowns. If `meth='median'` the function will use either the median (in case of numeric variables) or the most frequent value (in case of factors), of the neighbours to fill in the NAs. If `meth='weighAvg'` the function will use a weighted average of the values of the neighbours. The weights are given by `exp(-dist(k,x)` where `dist(k,x)` is the euclidean distance between the case with NAs (x) and the neighbour k.

Value

A dataframe with imputed values.

Note

This is a slightly modified function from package `DMwR` by Luis Torgo. The modification allows the units with missing values at almost all variables.

Author(s)

Luis Torgo

References

Torgo, L. (2010) Data Mining using R: learning with case studies, CRC Press (ISBN: 9781439810187).

See Also

`seqKNNimp`

Examples

```
mtcars$mpg[sample(1:nrow(mtcars), size = 5, replace = FALSE)] <- NA
KNNimp(data = mtcars)
```

ldaPlus *Linear discriminant analysis*

Description

The function performs a linear discriminant analysis (by using the MASS::lda function). Compared to the MASS::lda function, the ldaPlus function enable to consider the prior probabilities to predict the values of a categorical variable, it provides with predicted values and with (Jack-knife) classification table and also with statistical test of canonical correlations between the variable that represents groups and numeric variables.

Usage

```
ldaPlus(x, grouping, pred = TRUE, CV = TRUE, usePriorBetweenGroups = TRUE, ...)
```

Arguments

x	A data frame with values of numeric variables.
grouping	Categorical variable that defines groups.
pred	Whether to return the predicted values based on the model. Default is TRUE.
CV	Whether to do cross-validation in addition to "ordinary" analysis, default is TRUE.
usePriorBetweenGroups	Whether to use prior probabilities also in estimating the model (compared to only in prediction); default is TRUE.
...	Arguments passed to function MASS::lda.

Details

The specified prior is not taken into account when computing eigenvalues and all statistics based on them (everything in components eigModel and sigTest of the returned value).

Value

The following objects are also a part of what is returned by the MASS::lda function.

- prior - Prior probabilities of class membership taken to estimate the model (it can be estimated based on the sample data or it can be provided by a reseacher).
- counts - Number of units in each category of categorical variable taken to estimate the model.
- means - Group means.
- scaling - Matrix that transforms observations to discriminant functions, normalized so that within groups covariance matrix is spherical.
- lev - Levels (groups) of the categorical variable.
- svd - Singular values, that give the ratio of the between-group and within-group standard deviations on linear discriminant variables. Their squares are the canonical F-statistics.

- N - Number of observations used.
- call - the (matched) function call.

The additional following objects are generated by the multiUS::ldaPlus function.

- standCoefWithin - Standardized coefficients (within groups) of discriminant function.
- standCoefTotal - Standardized coefficients of discriminant function.
- betweenGroupsWeights - Proportions/priors used when estimating the model.
- sigTest - Test of canonical correlations between the variable that represent groups (binary variable) and numeric variables (see function testCC for more details) (Ho: The current and all the later canonical correlations equal to zero.).
- eigModel - Table with eigenvalues and canonical correlations (see function testCC for more details).
- centroids - Means of discriminant variables by levels of categorical variable (not predicted, but actual).
- corr - Pooled correlations within groups (correlations between values of numerical variables and values of linear discriminat function(s)).
- pred
  - class - Predicted values of categorical variable
  - posterior - Posterior probabilities (the values of the Fisher's calcification linear discrimination function)
  - x - Estimated values of discriminat function(s) for each unit
- class - Classification table:
  - orgTab - Frequency table.
  - perTab - Percentages.
  - corPer - Percentage of correctly predicted values (alternatively, percentage of correctly classified units).
- classCV - Similar to class but based on cross validation (Jack-knife).

Author(s)

Aleš Žiberna

References

R Data Analysis Examples: Canonical Correlation Analysis, UCLA: Statistical Consulting Group. From http://www.ats.ucla.edu/stat/r/dae/canonical.htm (accessed Decembar 27, 2013).

Examples

```
ldaPlus(x = mtcars[,c(1, 3, 4, 5, 6)], grouping = mtcars[,10])
```

makeFactorLabels *Make factor labels*

Description

The function transforms a numeric varibale into categorical one, based on the attribute data from a given SPSS file.

Usage

```
makeFactorLabels(x, reduce = TRUE, ...)
```

Arguments

x	Data for the selected variable, see Details.
reduce	Wheter to reduce categories with zero frequency, default is TRUE.
...	Arguments passed to function factor.

Details

Data have to be imported by using the foreign::read.spss function. The use of the function make sence when the parameter use.value.lables in the function read.spss is set to FALSE.

Value

Categorical variable (vector).

Author(s)

Aleš Žiberna

mapLda *LDA mapping*

Description

The function draws two dimensional map of discriminant functions.

Usage

```
mapLda(
  object,
  xlim = c(-2, 2),
  ylim = c(-2, 2),
  npoints = 101,
  prior = object$prior,
  dimen = 2,
  col = NULL
)
```

Arguments

object	Object obtained by ldaPlus function or MASS::lda function.
xlim	Limits of the x-axis.
ylim	Limits of the y-axis.
npoints	Number of points on y-axis and x-axis (i.e., drawing precision).
prior	Prior probabilities of class membership to estimate the model (they can be estimated based on the sample data or they can be provided by a reseacher).
dimen	Number of dimensions used for prediction. Probably only 2 (as these are used for drawing) makes sense.
col	Vector of mapping colors, default is NULL (i.e., it takes the default R colors).

Value

No return value, called for side effects (plotting a map).

Author(s)

Aleš Žiberna

Examples

```
# Estimate the LDA model:
ldaCars <- ldaPlus(x = mtcars[,c(1, 3, 4, 5, 6)], grouping = mtcars[,10])
# Plot LDA map:
mapLda(ldaCars)
```

Omega — *Simple version of omega coefficient - measure of measurement internal consistency based on factor analysis*

Description

The function omega coefficient, which is a measure of measurement internal consistency based on factor analysis, based on the covariance or correlation matrix. psych::fa is used to preform factor analysis.

Usage

```
Omega(
  C,
  fm = "ml",
  nfactors = 1,
  covar = TRUE,
  usePsych = TRUE,
  returnFaRes = FALSE,
  rotation = "none",
  ...
)
```

Arguments

C	Covariance or correlation matrix.
fm	Factor analysis method, maximum likelihood ("ml") by default. See psych::fa for details. Only used if usePsych is TRUE and psych package is available.
nfactors	Number of factors, 1 by default, psych::fa for details.
covar	Should the input C be treated as covariance matrix. Defaults to TRUE. If set to FALSE, the input C is converted to correlation matrix using stats::cov2cor.
usePsych	Should psych package or more precisely psych::fa be used to perform factor analysis. Defaults to TRUE. If FALSE or psych package is not available, stats::factanal is used.
returnFaRes	Should results of factor analysis be returned in addition to the computed omega coefficient. FALSE by default.
rotation	Rotation to be used in factor analysis. Defaults to "none", as it does not influence the Omega coefficient. Used only if returnFaRes is TRUE. Included if one wants to customize the results of factor analyisis. See psych::fa or stats::factanal for details (depending on which function is used, see usePsych).
...	Additional parameters to psych::fa or stats::factanal (depending on which function is used, see usePsych).

Value

By default just the value of the omega coefficient. If returnFaRes is TRUE, then a list with two elements:

- omega - The value of the omega coefficient.
- faRes - The result of factor analysis.

Author(s)

Ales Ziberna

Examples

```
Omega(C=cor(mtcars[,1:6]),nfactors=1)
Omega(C=cor(mtcars[,1:6]),nfactors=1,returnFaRes=TRUE)
```

plotCCA *Plot a solution of canonical correlations*

Description

It plots the canonical solution that is obtained by the function multiUS::cancorPlus.

Usage

```
plotCCA(
  ccRes,
  xTitle = "X",
  yTitle = "Y",
  inColors = TRUE,
  scaleLabelsFactor = 1/2,
  what = "reg",
  nDigits = 2,
  mar = c(1, 2, 1, 1)
)
```

Arguments

ccRes	The output of multiUS::cancorPlus.
xTitle	The title of the first set of variables.
yTitle	The title of the second set of variables.
inColors	Whether plot should be plotted in colours (TRUE) (default) or in black and white (FALSE).
scaleLabelsFactor	Parameter for setting the size of values (default is 1/2). The size of plotted values is proportional to its value to the power of scaleLabelsFactor.
what	Whether to plot regression coefficients ("reg") (default) or correlations (i.e., canonical structure loadings) ("cor").
nDigits	Number of decimal places.
mar	Margins, default is mar = c(1, 2, 1, 1), see graphics::par.

Value

It plots the plot.

Author(s)

Marjan Cugmas

Examples

```
tmp<-cancorPlus(x = mtcars[, c(1,2,3)], y = mtcars[, c(4,5, 6)], useCCApackage = TRUE)
plotCCA(tmp, scaleLabelsFactor = 1/2, what = "cor")
```

plotMeans *Plot the means*

Description

The function plots the means of several numerical variables by the levels of one categorical variable.

Usage

```
plotMeans(
  x,
  by,
  plotCI = TRUE,
  alpha = 0.05,
  ylab = "averages",
  xlab = "",
  plotLegend = TRUE,
  inset = 0.01,
  xleg = "topleft",
  legPar = list(),
  gap = 0,
  labels = NULL,
  ...
)
```

Arguments

x	Data frame with values of numeric variables.
by	Categorical variable that defines groups.
plotCI	Whether to plot confidence intervals or not, default is TRUE.
alpha	A confidence level for calculating confidence intervals (default is 0.05).
ylab	The title of y-axis.
xlab	The title of x-axis.
plotLegend	Whether to plot a legend or not, default is TRUE.
inset	Inset distance(s) from the margins as a fraction of the plot region when legend is placed by keyword.
xleg	Position of a legend, default is topleft.
legPar	Additional parameters for a legend. They have to be provided in a list format.
gap	Space left between the center of the error bar and the lines marking the error bar in units of the height (width). Defaults to 1.0
labels	Labels of x-axis.
...	Arguments passed to functions matplot and axis.

Value

A list with the following elements:

- means - mean values by groups.
- CI - widths of confidence intervals by groups.

Author(s)

Aleš Žiberna

Examples

```
plotMeans(x = mtcars[, c(1, 3, 5)], by = mtcars[,8])
```

`predict.ldaPlus` *Predict the values of a categorical variable based on a linear discriminant function*

Description

The function predicts the values of a categorical variable based on a linear discriminat function.

Usage

```
## S3 method for class 'ldaPlus'
predict(
  object,
  newdata,
  prior = object$prior,
  dimen,
  method = c("plug-in", "predictive", "debiased"),
  betweenGroupsWeights = object$betweenGroupsWeights,
  ...
)
```

Arguments

object	Object obtained by the ldaPlus function or by the MASS::lda.
newdata	New dataset (without categorical variable).
prior	Prior probabilities of class membership to be used to predict values.
dimen	The number of dimensions/linear discriminant functions to use. Defaults to all.
method	Possible values are plug-in, predictive and debiased.
betweenGroupsWeights	The proportions/weights used when computing the grand/total mean from group means.
...	other arguments passed to function MASS::predict.

Value

A list with the following elements:

- `class` - Predicted values of categorical variable.
- `posterior` - Posterior probabilities (the values of the Fisher's calsification linear discrimination function).
- `x` - Estimated values of discriminat function(s) for each unit.

Author(s)

Aleš Žiberna

See Also

`MASS::predict`

Examples

```
# Use the first 20 cars to estimate the model and the rest of cars to predict
# (for each car) wheter it has a V-shape engine or a straight engine.
ldaCars <- ldaPlus(x = mtcars[1:20,c(1, 2, 4, 5, 6)], grouping = mtcars[1:20,8])
predict.ldaPlus(object = ldaCars, newdata = mtcars[20:32,c(1, 2, 4, 5, 6)])
```

---

printP *Print p-value*

Description

The function round and prints p-value.

Usage

```
printP(p)
```

Arguments

`p`	Value to be printed.

Value

A string (formatted p-value).

Author(s)

Marjan Cugmas

Examples

```
printP(p = 0.523)
printP(p = 0.022)
printP(p = 0.099)
```

---

renameVar *Rename variables*

Description

The function for renaming one or several variables in a dataframe.

Usage

```
renameVar(data, renames)
```

Arguments

data	A dataframe.
renames	A list with oldnames and newnames (e.g, list("oldname1" = "newname1", "oldname2" = "newname2")).

Value

A dataframe with renamed columns.

Author(s)

Marjan Cugmas

Examples

```
renameVar(mtcars, list("cyl" = "Cylinders", "wt" = "Weight", "am" = "Transmission"))
```

---

seqKNNimp *Sequential KNN imputation method*

Description

This function estimates missing values sequentially from the units that has least missing rate, using weighted mean of k nearest neighbors.

Usage

```
seqKNNimp(data, k = 10)
```

Arguments

data	A data frame with the data set.
k	The number of nearest neighbours to use (defaults to 10).

Details

The function separates the dataset into an incomplete set with missing values and into a complete set without missing values. The values in an incomplete set are imputed in the order of the number of missing values. A missing value is filled by the weighted mean value of a corresponding column of the nearest neighbour units in the complete set. Once all missing values for a given unit are imputed, the unit is moved into the complete set and used for the imputation of the rest of units in the incomplete set. In this process, all missing values for one unit can be imputed simultaneously from the selected neighbour units in the complete set. This reduces execution time from previously developed KNN method that selects nearest neighbours for each imputation.

Value

A dataframe with imputed values.

Note

This is the function from package SeqKNN by Ki-Yeol Kim and Gwan-Su Yi.

Author(s)

Ki-Yeol Kim and Gwan-Su Yi

References

Ki-Yeol Kim, Byoung-Jin Kim, Gwan-Su Yi (2004.Oct.26) "Reuse of imputed data in microarray analysis increases imputation efficiency", BMC Bioinformatics 5:160.

See Also

KNNimp

Examples

```
mtcars$mpg[sample(1:nrow(mtcars), size = 5, replace = FALSE)] <- NA
seqKNNimp(data = mtcars)
```

`small2other` *Recoding the smallest categories to "other" value in case of too many or too small categories.*

Description

The smallest categories are recoded to "other" or user specified string. The variables is converted to factor if not already.

Usage

```
small2other(
  x,
  maxLevels = 12,
  minFreq = 0,
  otherValue = "other",
  convertNA = TRUE,
  orderLevels = FALSE,
  otherLast = FALSE
)
```

Arguments

`x`	The variable to be recoded.
`maxLevels`	The maximum number of levels after recoding
`minFreq`	The minimal frequency after recoding.
`otherValue`	The name give to the new category
`convertNA`	Should the `NA` values be converted to ordinary values. If `TRUE`, they are converted to string `"NA"`. If `FALSE`, there are left as missing and ignored in the recording.
`orderLevels`	How should the categories be ordered. Possible values are: • `FALSE` - do not change the ordering (default) • `alpha` - alphabetically; and • freq - based on frequencies (highest frequencies first).
`otherLast`	Only used if category with `otherValue` was created. If `TRUE`, the `otherValue` is placed as last category regardless of the `orderLevels` argument. Defaults to `FALSE`.

testCC *Test of canonical correlations*

Description

The function perform the Wilk's test for the statistical significance of canonical correlations.

Usage

```
testCC(cor, n, p, q)
```

Arguments

cor	Vector with canonical correlations.
n	Number of units.
p	Number of variables in the first group of variables.
q	Number of variables in the second group of variables.

Value

The results are organized in a `list` format with two data tables:

`sigTest`

- `WilksL` - Value of the Wilk's lambda statistic (it is a generalization of the multivariate R2; values near 0 indicate high correlation while values near 1 indicate low correlation).
- `F` - Corresponding (to Wilk's lambda) F-ratio.
- `df1` - Degrees of freedom for the corresponding F-ratio.
- `df2` - Degrees of freedom for the corresponding F-ratio.
- `p` - Probability value (p-value) for the corresponding F-ratio (Ho: The current and all the later canonical correlations equal to zero).

`eigModel`

- `Eigenvalues` - Eigenvalues of the canonical roots.
- `%` - Proportion of explained variance of correlation.
- `Cum %` - Cumulative proportion of explained variance of correlation.
- `Cor` - Canonical correlation coeficient.
- `Sq. Cor` - Squared canonical correlation coeficient.

Author(s)

Aleš Žiberna

References

R Data Analysis Examples: Canonical Correlation Analysis, UCLA: Statistical Consulting Group. From http://www.ats.ucla.edu/stat/r/dae/canonical.htm (accessed Decembar 27, 2013).

Examples

```
testCC(cor = c(0.76, 0.51, 0.35, 0.28, 0.10), n = 51, p = 5, q = 5)
```

`Theta` *Theta coefficient - measure of measurement internal consistency based on principal component analysis*

Description

The function theta coefficient, which is a measure of measurement internal consistency based on principal component analysis, or more precisely first eigenvalue.

Usage

```
Theta(C)
```

Arguments

`C`	Covariance or correlation matrix.

Value

The value of the theta coefficient.

Author(s)

Ales Ziberna

Examples

```
Theta(C=cor(mtcars[,1:6]))
```

wardKF *Calculate the value of the Ward criterion function*

Description

The function calculate the value of the Ward criterion function, based on a set of numerical variables and one categorical variable (partition).

Usage

```
wardKF(X, clu)

wardCF(X, clu)
```

Arguments

X	Data frame with values of numerical variables (usually the ones that were/are used for clustering).
clu	Partition.

Value

The value of the Ward criterion function.

Author(s)

Aleš Žiberna

Index