

Package ‘scmamp’

October 21, 2016

Type Package

Title Statistical Comparison of Multiple Algorithms in Multiple Problems

Version 0.2.55

Date 2016-10-20

LazyData yes

Suggests knitr, rmarkdown

Imports utils, graphics, ggplot2, graph, reshape2, Rgraphviz, methods

VignetteBuilder knitr

Description Given a matrix with results of different algorithms for different problems, the package uses statistical tests and corrections to assess the differences between algorithms.

License GPL-2

BugReports <https://github.com/b0rxa/scmamp/issues>

Collate tests.R post_hoc.R plotting.R scmamp.R latex.R
loading_functions.R data_manipulation.R

RoxygenNote 5.0.1

NeedsCompilation no

Author Borja Calvo [aut, cre],
Guzman Santafe [aut]

Maintainer Borja Calvo <borja.calvo@ehu.eus>

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2016-10-21 12:28:46

R topics documented:

adjustBergmannHommel	3
adjustFinner	4

adjustHolland	5
adjustLi	6
adjustRom	7
adjustShaffer	8
anovaTest	9
booleanMatrix	9
contrastEstimationMatrix	10
customPost	11
data.blum.2015	12
data.gh.2008	12
data.gh.2010	13
drawAlgorithmGraph	13
exhaustiveSets	14
filterData	15
friedmanAlignedRanksPost	16
friedmanAlignedRanksTest	17
friedmanPost	18
friedmanTest	19
imanDavenportTest	20
multipleComparisonTest	21
nemenyiTest	22
plotCD	23
plotDensities	24
plotPvalues	25
plotRanking	25
postHocTest	26
qqplotGaussian	29
quadePost	30
quadeTest	31
rankMatrix	32
readComparisonDir	32
readComparisonFile	33
readExperimentDir	34
readExperimentFile	36
scmamp	37
summarizeData	37
tukeyPost	38
tukeyTest	39
wilcoxonSignedTest	40
writeTabular	41

adjustBergmannHommel *Bergmann and Hommel dynamic correction of p-values.*

Description

This function takes the particular list of possible hypothesis to correct for multiple testing, as defined in Bergmann and Hommel (1994).

Usage

```
adjustBergmannHommel(raw.matrix)
```

Arguments

`raw.matrix` Raw p-values in a matrix.

Details

The test has been implemented according to the version in Garcia and Herrera (2008), page 2680-2682.

Value

A matrix with the corrected p-values

References

S. Garcia and F. Herrera (2008) An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for All Pairwise Comparisons. *Journal of Machine Learning Research*, 9, 2677-2694.

G. Bergmann and G. Hommel (1988) Improvements of general multiple test procedures for redundant systems of hypotheses. In P. Bauer, G. Hommel and E. Sonnemann, editors, *Multiple Hypotheses Testing*, 100-115, Springer, Berlin.

Examples

```
data(data_gh_2008)
raw.pvalues <- friedmanAlignedRanksPost(data.gh.2008)
raw.pvalues
adjustBergmannHommel (raw.pvalues)
```

`adjustFinner`*Finner correction of p-values*

Description

This function takes the particular list of possible hypothesis to correct for multiple testing, as defined in Finner (1993)

Usage

```
adjustFinner(pvalues)
```

Arguments

`pvalues` Raw p-values in either a vector or a matrix. Note that in case the p-values are in a matrix, all the values are used for the correction. Therefore, if the matrix contains repeated values (as those output by some methods in this package), the repetitions have to be removed.

Details

The test has been implemented according to the version in Garcia *et al.* (2010), page 2680-2682.

Value

A vector or matrix with the corrected p-values

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparison in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

H. Finner (1993) On a monotocity problem in ste-down mulitple test procedures. *Journal of the American Statistical Association*, 88, 920-923.

Examples

```
data(data_gh_2008)
raw.pvalues <- friedmanPost(data.gh.2008)
raw.pvalues
adjustFinner (raw.pvalues)
```

adjustHolland	<i>Holland correction of p-values.</i>
---------------	----------------------------------------

Description

This function takes the particular list of possible hypothesis to correct for multiple testing, as defined in Holland and Copenhaver (1987)

Usage

```
adjustHolland(pvalues)
```

Arguments

pvalues	Raw p-values in either a vector or a matrix. Note that in case the p-values are in a matrix, all the values are used for the correction. Therefore, if the matrix contains repeated values (as those output by some methods in this package), the repetitions have to be removed.
---------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Details

The test has been implemented according to the version in Garcia *et al.* (2010), page 2680-2682.

Value

A vector or matrix with the corrected p-values

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera, F. (2010) Advanced nonparametric tests for multiple comparison in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

B. S. Holland and M. D. Copenhaver (1987) An improved sequentially rejective Bonferroni test procedure *Biometrics*, 43, 417-423.

Examples

```
data(data_gh_2008)
raw.pvalues <- friedmanPost(data.gh.2008)
raw.pvalues
adjustHolland (raw.pvalues)
```

`adjustLi`*Li correction of p-values.*

Description

This function takes the particular list of possible hypothesis to correct for multiple testing, as defined in Li (2008).

Usage

```
adjustLi(pvalues)
```

Arguments

`pvalues` Raw p-values in either a vector or a matrix. Note that in case the p-values are in a matrix, all the values are used for the correction. Therefore, if the matrix contains repeated values (as those output by some methods in this package), the repetitions have to be removed.

Details

The test has been implemented according to the version in Garcia *et al.* (2010), page 2680-2682. This is a simple procedure that provides good results when the highest p-value corrected is below 0.5. However, when the highest p-value is close to 1 the correction is extremely conservative. Actually, when the highest p-value is 1, all the corrected p-values are set at 1. Therefore, it is not advisable to be used under these circumstances.

Value

A vector or matrix with the corrected p-values

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparison in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

J. Li (2008) A two-step rejection procedure for testing multiple hypotheses. *Journal of Statistical Planning and Inference*, 138, 1521-1527.

Examples

```
data(data_gh_2008)
raw.pvalues <- friedmanPost(data.gh.2008)
adjustLi(raw.pvalues)
```

`adjustRom`*Rom correction of p-values*

Description

This function takes the particular list of possible hypothesis to correct for multiple testing, as defined in Rom (1990)

Usage

```
adjustRom(pvalues, alpha = 0.05)
```

Arguments

<code>pvalues</code>	Raw p-values in either a vector or a matrix. Note that in case the p-values are in a matrix, all the values are used for the correction. Therefore, if the matrix contains repeated values (as those output by some methods in this package), the repetitions have to be removed.
<code>alpha</code>	value for the overall test

Details

The test has been implemented according to the version in Garcia *et al.* (2010), page 2680-2682.

Value

A vector or matrix with the corrected p-values

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparison in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

D. M. Rom (1990) A sequentially rejective test procedure based on a modified Bonferroni inequality. *Biometrika*, 77, 663-665.

Examples

```
data(data_gh_2008)
raw.pvalues <- friedmanPost(data.gh.2008)
raw.pvalues
adjustRom(raw.pvalues, alpha=0.05)
```

adjustShaffer	<i>Shaffer's correction of p-values in pairwise comparisons.</i> .
---------------	--------------------------------------------------------------------

Description

This function implements the Shaffer's (static) multiple testing correction when the p-values correspond with pairwise comparisons.

Usage

```
adjustShaffer(raw.matrix)
```

Arguments

`raw.matrix` A matrix with the pairwise p-values. The p-values have to be, at least, in the upper part of the matrix.

Details

The test has been implemented according to the version in Garcia and Herrera (2008), page 2680.

Value

A symmetric matrix with the corrected p-values.

References

S. Garcia and F. Herrera (2008) An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for All Pairwise Comparisons. *Journal of Machine Learning Research*, 9, 2677-2694.

J.P. Shaffer (1986) Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, 81(395), 826-831.

Examples

```
data(data_gh_2008)
raw.pvalues <- friedmanPost(data.gh.2008)
raw.pvalues
adjustShaffer(raw.pvalues)
```

anovaTest	<i>ANOVA test for multiple comparisons</i>
-----------	--------------------------------------------

Description

This function performs F-test for K populations means

Usage

```
anovaTest(data, ...)
```

Arguments

data	Matrix where the test is performed
...	Ignored

Details

The test has been implemented according to Test 22 in Kanji (2006).

Value

A list with class "htest" containing the following components: `statistic`, the value of the statistic used in the test; `p.value`, the p-value for the test; `method`, a character string indicating what type of test was performed and `data.name`, a character string giving the name of the data.

References

Kanji, G. K. (2006) *100 Statistical Tests*. SAGE Publications Ltd, 3rd edition.

Examples

```
data(data_gh_2008)
anovaTest(data.gh.2008)
```

booleanMatrix	<i>Creation of boolean matrices for highlighting results</i>
---------------	--------------------------------------------------------------

Description

A simple function to create boolean matrices to be used when constructing LaTeX tables.

Usage

```
booleanMatrix(data, find = "max", th = 0, by = "row")
```

Arguments

data	It can be a data frame, a matrix or a vector.
find	A string indicating what has to be detected. Possible values are: <ul style="list-style-type: none"> • 'eq' All values equal to the value passed in th • 'le' All values lower or equal to the value passed in th • 'ge' All values greater or equal to the value passed in th • 'lw' All values lower than the value passed in th • 'gt' All values greater than the value passed in th • 'min' Minimum value in each row / column / matrix • 'max' Maximum value in each row / column / matrix
th	Thershold used when find is set to 'eq', 'ge', 'le', 'gt' or 'lw'.
by	A string or string vector indicating where the min/max values have to be find. It can be 'row', 'col' or 'mat' for the row, column and matrix min/max respectively.

Value

A boolean matrix that matches in dimension the output data and where the identified elements are marked as TRUE.

Examples

```
data('data_gh_2008')
booleanMatrix(data.gh.2008, find='min', by='row')
booleanMatrix(data.gh.2008, find='ge', th=0.5)
```

contrastEstimationMatrix

Contrast estimation based on medians

Description

This function performs estimates the contrast between algorithms through the medians

Usage

```
contrastEstimationMatrix(data)
```

Arguments

data	Matrix or data frame with the data to compare
------	-----------------------------------------------

Details

The test has been implemented according to Garcia *et al.* (2010), Section 3.3.

Value

A matrix where the estimation of all the pairs of differences are output. The differences correspond to row-column.

References

Kanji, G. K. (2006) *100 Statistical Tests*. SAGE Publications Ltd, 3rd edition.

Examples

```
data(data_gh_2008)
contrastEstimationMatrix(data.gh.2008)
```

customPost

Function to use custom tests to perform post hoc comparisons.

Description

This function computes the raw p-values for all vs. all or all vs. control comparisons using a custom function.

Usage

```
customPost(data, control = NULL, test, ...)
```

Arguments

data	Data set (matrix or data.frame) to apply the test. The column names are taken as the groups and the values in the matrix are the samples.
control	Either the number or the name of the column for the control algorithm. If this parameter is not provided, the all vs all comparison is performed.
test	Function to perform the test. It requires two parameters, x and y, the two samples to be compared, and it has to return a list that contains, at least, one element called p.value (as the htest objects that are usually returned by R's statistical test implementations).
...	Additional parameters for the test function.

Value

A matrix with all the pairwise raw p-values.

Examples

```
data(data_gh_2008)
test <- function(x, y, ...) {
  t.test(x, y, paired=TRUE)
}
customPost(data.gh.2008, control=1, test=test)
customPost(data.gh.2008, test=test)
```

data.blum.2015

Comparison of optimization algorithms in Blum et al. (2015)

Description

This dataset contains part of the results obtained in the comparison of decentralized optimization algorithms presented in Blum *et al.* (2015). The dataset contains 900 rows and 10 columns. Each row represents an instance of the maximum independent set problem (a graph). The first two are descriptors of the problem in each row (size and radius used to create random geometric graphs) and the other 8 contain the results obtained by 8 algorithms for the MIS problem instance.

Format

A data frame with 10 columns and 900 rows

Source

C. Blum, B. Calvo and M.J. Blesa (2015) FrogCOL and FrogMIS: New Decentralized Algorithms for Finding Large Independent Sets in Graphs. *Swarm Intelligence*. In press.

data.gh.2008

Example in Garcia and Herrera (2008)

Description

Dataset corresponding to the accuracy of 5 classifiers in 30 datasets. Each algorithm is in a column. This is the dataset used as example in Garcia and Herrera (2008).

Format

A data frame with 5 columns and 30 rows

Source

S. Garcia and F. Herrera (2008) An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research*. 9, 2677-2694.

 data.gh.2010

Example in Garcia and Herrera (2010)

Description

Dataset corresponding to the accuracy of 4 classifiers in 24 datasets. Each algorithm is in a column. This is the dataset used as example in Garcia and Herrera (2010).

Format

A data frame with 4 columns and 24 rows

Source

S. Garcia and F. Herrera (2010) Advanced Nonparametric Tests for Multiple Comparison in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. *Information Sciences*, 180, 2044-2064.

 drawAlgorithmGraph

Hypotheses represented as a graph

Description

This function can be used to plot a graph where algorithms are nodes and algorithms that cannot be regarded as different are joined by an edge.

Usage

```
drawAlgorithmGraph(pvalue.matrix, mean.value, ..., alpha = 0.05,
  font.size = 15, highlight = "min", highlight.color = "chartreuse3",
  node.color = "gray30", font.color = "white", digits = 2,
  node.width = 5, node.height = 2)
```

Arguments

pvalue.matrix	Matrix with the p-values
mean.value	Vector of values to be written together with the name of the algorithm
...	Additional parameters to the Rgraphviz function. This is mainly to change the layout of the graph
alpha	Significance level to determine which hypotheses are rejected.
font.size	Size of the font for the node labels.
highlight	A character indicating which node has to be highlighted. It can be the one with the maximum value ('max'), the minimum value ('min') or none ('none').

highlight.color	Any R valid color for the highlighted node.
node.color	Any R valid color for the non-highlighted nodes.
font.color	Any R valid color for the node labels.
digits	Number of digits to display the value associated to each node
node.width	Numeric value for the width of the node
node.height	Numeric value for the height of the node

See Also

[plotPvalues](#), [plotRanking](#), [plotCD](#)

Examples

```
data(data_blum_2015)
data <- filterData(data.blum.2015, condition="Size == 1000", remove.cols=1:2)
res <- postHocTest(data, test = "friedman", use.rank=TRUE, correct="bergmann")
## This function requires the package Rgraphviz
# drawAlgorithmGraph(res$corrected.pval, res$summary)
```

exhaustiveSets	<i>Complete set of exhaustive sets.</i>
----------------	-----------------------------------------

Description

This function implements the algorithm in Figure 1, Garcia and Herrera (2008) to create, given a set, the complete set of exhaustive sets E.

Usage

```
exhaustiveSets(set)
```

Arguments

set	Set to create the exhaustive sets. The complexity of this algorithm is huge, so use with caution for sets of more than 7-8 elements. Indeed, the implementation, as it is, can be hardly used from sizes beyond 9.
-----	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Details

The algorithm makes use of ‘exhaustive.sets’, a structure provided with the package that contains the precomputed sets for size up to 9. With this structure the exhaustive sets are generated immediately, but if the data is, for some reason, not loaded, the computation may take several hours (or even days, depending on the size of the set).

Value

A list with all the possible exhaustive sets, without repetitions.

Examples

```
exhaustiveSets(c("A", "B", "C", "D"))
```

filterData	<i>Expression based row filtering</i>
------------	---------------------------------------

Description

This is a simple function to filter data based on an expression defined using the column names

Usage

```
filterData(data, condition = "TRUE", remove.cols = NULL)
```

Arguments

data	A NAMED matrix or data frame to be filtered (column names are required).
condition	A string indicating the condition that the row have to fulfill to be retained. The column names are used as variables in the condition (see examples bellow).
remove.cols	Either a vector of column names or a vector of column indices to be removed from the result

Value

The original data where the rows for which the condition is FALSE and the columns in the vector remove.cols have been removed

See Also

[summarizeData](#), [writeTabular](#) and the vignette `vignette(topic="Data_loading_and_manipulation", package="scma")`

Examples

```
data(data_gh_2008)
names(data.gh.2008)
filterData(data.gh.2008, condition="CN2 > 0.7 & Kernel < 0.7", remove.cols=1:2)
```

friedmanAlignedRanksPost

Friedman's Aligned Ranks post hoc raw p-values

Description

This function computes the raw p-values for the post hoc based on Friedman's Aligned Ranks test.

Usage

```
friedmanAlignedRanksPost(data, control = NULL, ...)
```

Arguments

data	Data set (matrix or data.frame) to apply the test. The column names are taken as the groups and the values in the matrix are the samples.
control	Either the number or the name of the column for the control algorithm. If this parameter is not provided, the all vs all comparison is performed.
...	Not used.

Details

The test has been implemented according to the version in Garcia *et al.* (2010), pages 2051,2054

Value

A matrix with all the pairwise raw p-values (all vs. all or all vs. control).

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

Examples

```
data(data_gh_2008)
friedmanAlignedRanksPost(data.gh.2008)
friedmanAlignedRanksPost(data.gh.2008, control=1)
```

`friedmanAlignedRanksTest`*Friedman's Aligned Ranks test*

Description

This function performs Friedman's Aligned Rank test for multiple comparisons

Usage

```
friedmanAlignedRanksTest(data, ...)
```

Arguments

<code>data</code>	Matrix where the test is performed
<code>...</code>	Ignored

Details

The test has been implemented according to the version in Garcia *et al.* (2008).

Value

A list with class "hstest" containing the following components: `statistic`, the value of the statistic used in the test; `parameter`, the two degrees of freedom of the F distribution; `p.value`, the p-value for the test; `method`, a character string indicating what type of test was performed and `data.name`, a character string giving the name of the data.

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

Examples

```
data(data_gh_2008)
friedmanTest(data.gh.2008)
```

friedmanPost	<i>Friedman's post hoc raw p-values</i>
--------------	-----------------------------------------

Description

This function computes the raw p-values for the post hoc based on Friedman's test.

Usage

```
friedmanPost(data, control = NULL, ...)
```

Arguments

data	Data set (matrix or data.frame) to apply the test. The column names are taken as the groups and the values in the matrix are the samples.
control	Either the number or the name of the column for the control algorithm. If this parameter is not provided, the all vs all comparison is performed.
...	Not used.

Details

The test has been implemented according to the version in Demsar (2006), page 12.

Value

A matrix with all the pairwise raw p-values (all vs. all or all vs. control).

References

J. Demsar (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

Examples

```
data(data_gh_2008)
friedmanPost(data.gh.2008)
friedmanPost(data.gh.2008, control=1)
```

friedmanTest	<i>Friedman's test</i>
--------------	------------------------

Description

This function performs Friedman's test for multiple comparisons

Usage

```
friedmanTest(data, ...)
```

Arguments

data	Matrix where the test is performed
...	Ignored

Details

The test has been implemented according to the version in Demsar (2006), page 11

Value

A list with class "hstest" containing the following components: `statistic`, the value of the statistic used in the test; `parameter`, the two degrees of freedom of the F distribution; `p.value`, the p-value for the test; `method`, a character string indicating what type of test was performed and `data.name`, a character string giving the name of the data.

References

Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

Examples

```
data(data_gh_2008)
friedmanTest(data.gh.2008)
```

imanDavenportTest	<i>Iman Davenport's modification of Friedman's test</i>
-------------------	---------------------------------------------------------

Description

This function performs Iman-Davenport modification of Friedman's test

Usage

```
imanDavenportTest(data, ...)
```

Arguments

data	Matrix where the test is performed
...	Ignored

Details

The test has been implemented according to the version in Demsar (2006), page 11

Value

A list with class "hstest" containing the following components: `statistic`, the value of the statistic used in the test; `parameter`, the two degrees of freedom of the F distribution; `p.value`, the p-value for the test; `method`, a character string indicating what type of test was performed and `data.name`, a character string giving the name of the data.

References

Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

Examples

```
data(data_gh_2008)
imanDavenportTest(data.gh.2008)
```

multipleComparisonTest

Tests for multiple comparisons

Description

This function is a wrapper to multiple comparison tests.

Usage

```
multipleComparisonTest(data, algorithms = NULL, group.by = NULL,
  test = "aligned ranks", correct = "finer", alpha = 0.05, ...)
```

Arguments

- | | |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data | A matrix or data frame containing the results obtained by the algorithms (columns) in each problem (rows). It can contain additional columns, but if any of the column has to be discarded (not used neither to group the problems nor to be part of the comparison), then it is mandatory to indicate, in the <code>algorithms</code> parameter, which columns contain the algorithm information. |
| algorithms | Vector with either the names or the indices of the columns that contain the values to be tested. If not provided, the function assumes that all the columns except those indicated in <code>group.by</code> represent the results obtained by an algorithm. |
| group.by | Vector with either the names or the indices of the columns to be used to group the data. Each group is tested independently. If <code>NULL</code> , all the data is used for a single comparison. |
| test | <p>Parameter that indicates the statistical test to be used. It can be either a string indicating one of the available test or a function. As a string, it can take the following values:</p> <ul style="list-style-type: none"> • 'friedman' - Friedman test, as in Garcia and Herrera (2010) • 'aligned ranks' Friedman's Aligned Ranks test, as in Garcia and Herrera (2010) • 'quade' - Quade test, as in Garcia and Herrera (2010) • 'anova' - ANOVA test, as in Test 22 in Kanji (2006). <p>If a function is provided, then it has to have as first argument a matrix containing the columns to be compared. The function has to return a list with, at least, an element named <code>p.value</code> (as the <code>htest</code> objects that are usually returned by R's test implementations).</p> |
| correct | <p>Either string indicating the type of correction that has to be applied or a function to correct the p-values for multiple testing; This parameter is only need in case the data is grouped. As a string, the valid values are:</p> <ul style="list-style-type: none"> • <code>holland</code> - Holland's procedure, as in Garcia and Herrera (2010) • <code>finer</code> - Finer's procedure, as in Garcia and Herrera (2010) • <code>rom</code> - Rom's procedure, as in Garcia and Herrera (2010) |

- `li` - Li's procedure, as in Garcia and Herrera (2010)
 - Any of the methods implemented in the `p.adjust` function. For a list of options, type `p.adjust.methods`
- . If a function is provided, the it has to receive, as first argument, a vector of p-values to be corrected and has to return a vector with the corrected p-values *in the same order* as the input vector.
- `alpha` Alpha value used in Rom's correction. By default, set at 0.05.
- `...` Special argument used to pass additional parameters to the statistical test and the correction method.

Value

In case the `group.by` argument is not provided (or it is `NULL`), the function return an object of class `hstest`. If columns for grouping are provided, then the function returns a matrix that includes, for each group, the values of the `group.by` columns, the raw p-value and the corrected p-value.

#' @seealso [friedmanTest](#), [friedmanAlignedRanksTest](#), [quadeTest](#), [anovaTest](#), [adjustShaffer](#), [adjustBergmannHommel](#), [adjustHolland](#), [adjustFinner](#), [adjustRom](#), [adjustLi](#)

References

S. Garcia and F. Herrera (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

Kanji, G. K. (2006) *100 Statistical Tests*. SAGE Publications Ltd, 3rd edition.

Examples

```
# Grouped data
data(data_blum_2015)
multipleComparisonTest (data=data.blum.2015,
  algorithms=c("FrogCOL", "FrogMIS", "FruitFly"),
  group.by=c("Size", "Radius"),
  test="quade", correct="finner")

# Not grouped data
data(data_gh_2008)
multipleComparisonTest (data=data.gh.2008, test="aligned ranks",
  correct="hochberg")
```

nemenyiTest

Nemenyi test

Description

This function performs the Nemenyi test

Usage

```
nemenyiTest(data, alpha = 0.05)
```

Arguments

data	Matrix or data frame where each algorithm is in a column
alpha	Significance level

Details

The test has been implemented according to the version in Demsar (2006), page 7

Value

A list with class "htest" containing the following components: `statistic`, the value of the statistic used in the test; `method`, a character string indicating what type of test was performed; `data.name`, a character string giving the name of the data and `diff.matrix`, a matrix with all the pairwise differences of average rankings

References

Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

Examples

```
data(data_gh_2008)
res <- nemenyiTest(data.gh.2008, alpha = 0.1)
res
res$diff.matrix
```

plotCD	<i>Critical difference plot</i>
--------	---------------------------------

Description

This function plots the critical difference plots shown in Demsar (2006)

Usage

```
plotCD(results.matrix, alpha = 0.05, cex = 0.75, ...)
```

Arguments

results.matrix	Matrix or data frame with the results for each algorithm
alpha	Significance level to get the critical difference. By default this value is 0.05
cex	Numeric value to control the size of the font. By default it is set at 0.75.
...	Additional arguments for rankMatrix

References

Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

See Also

[drawAlgorithmGraph](#), [plotRanking](#), [plotPvalues](#)

Examples

```
data(data_gh_2008)
plotCD(data.gh.2008, alpha=0.01)
```

plotDensities

Kernel based density estimation of the samples

Description

This function estimates and plots the densities of the results of each algorithm

Usage

```
plotDensities(data, ...)
```

Arguments

data	A matrix where columns represent the algorithms
...	The plot is created using <code>ggplot2</code> . This special parameter can be used to pass additional parameters to the <code>geom_line</code> function used to plot the sample points. It can also be used to pass additional arguments to the <code>density</code> function, which is used to estimate the densities.

Value

A `ggplot` object.

See Also

[qqplotGaussian](#)

Examples

```
data(data_gh_2010)
plotDensities(data.gh.2010)
```

plotPvalues	<i>Plotting the p-value matrix</i>
-------------	------------------------------------

Description

This function plots the p-value matrix as a tile plot.

Usage

```
plotPvalues(pvalue.matrix, alg.order = NULL, show.pvalue = TRUE,  
            font.size = 5)
```

Arguments

pvalue.matrix	Matrix with the p-values to plot
alg.order	A permutation indicating the reordering of the algorithms
show.pvalue	Logical value indicating whether the numerical values have to be printed
font.size	Size of the p-values, if printed

Value

A [ggplot](#) object.

See Also

[drawAlgorithmGraph](#), [plotCD](#)

Examples

```
data(data_gh_2008)  
pvalues <- friedmanPost(data.gh.2008)  
ordering <- order(summarizeData(data.gh.2008))  
plotPvalues(pvalues, alg.order=ordering)
```

plotRanking	<i>Ranking Plots</i>
-------------	----------------------

Description

This function creates a plot similar to the critical difference plot, but applicable to any corrected pvalue.

Usage

```
plotRanking(pvalues, summary, alpha = 0.05, cex = 0.75,
            decreasing = FALSE)
```

Arguments

pvalues	Matrix or data frame with the p-values used to determine the differences
summary	Summary values used to place the algorithms. Typically it will be the average ranking, but it can be any other value
alpha	Significance level to determine the significativity of the differences. By default this value is 0.05
cex	Numeric value to control the size of the font. By default it is set at 0.75.
decreasing	A logical value to determine whether the values have to be plotter from smaller to larger or the other way round.

See Also

[drawAlgorithmGraph](#), [plotCD](#), [plotPvalues](#)

Examples

```
data(data_gh_2008)
test <- postHocTest(data.gh.2008, test="friedman", correct="bergmann", use.rank=TRUE)
plotRanking(pvalues=test$corrected.pval, summary=test$summary, alpha=0.05)
```

postHocTest

Post hoc tests for multiple comparison analyses

Description

This function is a wrapper to run the post hoc tests. It can run both all vs. control and all vs. all post hoc tests.

Usage

```
postHocTest(data, algorithms = NULL, group.by = NULL, test = "friedman",
            control = NULL, use.rank = FALSE, sum.fun = mean, correct = "finer",
            alpha = 0.05, ...)
```

Arguments

<code>data</code>	A matrix or data frame containing the results obtained by the algorithms (columns) in each problem (rows). It can contain additional columns, but if any of the column has to be discarded (not used neither to group the problems nor to be part of the comparison), then it is mandatory to indicate, in the <code>algorithms</code> parameter, which columns contain the algorithm information.
<code>algorithms</code>	Vector with either the names or the indices of the columns that contain the values to be tested. If not provided, the function assumes that all the columns except those indicated in <code>group.by</code> represent the results obtained by an algorithm.
<code>group.by</code>	Vector with either the names or the indices of the columns to be used to group the data. Each group is tested independently. If NULL, all the data is used for a single comparison.
<code>test</code>	<p>Parameter that indicates the statistical test to be used. It can be either a string indicating one of the available test or a function. As a string, it can take the following values:</p> <ul style="list-style-type: none"> • <code>'wilcoxon'</code> - Wilcoxon Signed Rank test, as in Demsar (2006) • <code>'t-test'</code> - t-test (R's <code>t.test</code> function with paired option set at TRUE) • <code>'friedman'</code> - Friedman post hoc test, as in Demsar (2006) • <code>'aligned_ranks'</code> - Friedman's Aligned Ranks post hoc test, as in Garcia and Herrera (2010) • <code>'quade'</code> - Quade post hoc test, as in Garcia and Herrera (2010) • <code>'tukey'</code> - Tukey's ANOVA post hoc test, as in Test 28 in Kanji (2006). <p>If a function is provided, then it has to have as first argument a matrix containing the columns to be compared. The function has to return a list with, at least, an element named <code>p.value</code> (as the <code>htest</code> objects that are usually returned by R's test implementations).</p>
<code>control</code>	Either the name or the index of a column in the dataset (one of those in the <code>algorithms</code> vector), to be used as control. Alternatively, this argument can be <code>'min'</code> , to select the algorithm with the minimum value, <code>'max'</code> , to select the algorithm with the maximum value as control. If the argument is not provided (or is NULL), all the pairwise comparisons are performed instead of all vs. control comparisons.
<code>use.rank</code>	If TRUE, then the summarization of the data is based on the ranks, rather than on the actual values. The selection of the algorithm with the maximum or minimum value is also done in terms of the summarized ranking.
<code>sum.fun</code>	Function to be used to summarize the data. By default, average is used.
<code>correct</code>	<p>Either string indicating the type of correction that has to be applied or a function to correct the p-values for multiple testing; This parameter is only need in case the data is grouped. As a string, the valid values are:</p> <ul style="list-style-type: none"> • <code>shaffer</code> - Shaffer's (static) procedure, as in Garcia and Herrera (2008) • <code>bergmann</code> - Bergman and Hommel's procedure (similar to Shaffer dynamic), as in Garcia and Herrera (2008) • <code>holland</code> - Holland's procedure, as in Garcia and Herrera (2010) • <code>finner</code> - Finner's procedure, as in Garcia and Herrera (2010)

- rom - Rom's procedure, as in Garcia and Herrera (2010)
 - li - Li's procedure, as in Garcia and Herrera (2010)
 - Any of the methods implemented in the `p.adjust` function. For a list of options, type `p.adjust.methods`
- . If a function is provided, the it has to receive, as first argument, a vector of p-values to be corrected and has to return a vector with the corrected p-values *in the same order* as the input vector.
- alpha Alpha value used in Rom's correction. By default, it is set at 0.05.
- ... Special argument used to pass additional parameters to the statistical test and the correction method.

Value

In all cases the function returns a list with three elements, the summarization of the data (a row per group), the raw p-values and the corrected p-values. When the data is grouped and all the pairwise comparisons are performed (no control is provided), the p-values are in three dimensional arrays where the last dimension is corresponds to the group. In any other cases the result is a matrix with one or more rows.

Note that Shaffer and Bergmann and Hommel's correction can only be applied when all the pairwise tests are conducted, due to their assumptions. Moreover, its use when the data is grouped (multiple pairwise comparisons) is not trivial and, thus, it is not possible to use it when the data is grouped.

References

S. Garcia and F. Herrera (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

Garcia S. and Herrera, F. (2008) An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for All Pairwise Comparisons. *Journal of Machine Learning Research*, 9, 2677-2694.

Kanji, G. K. (2006) *100 Statistical Tests*. SAGE Publications Ltd, 3rd edition.

Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

See Also

[friedmanPost](#), [friedmanAlignedRanksPost](#), [quadePost](#), [tukeyPost](#), [adjustShaffer](#), [adjustBergmannHommel](#), [adjustHolland](#), [adjustFinner](#), [adjustRom](#), [adjustLi](#)

Examples

```
# Grouped data, all pairwise
data(data_blum_2015)
res <- postHocTest (data=data.blum.2015, algorithms=c("FrogCOL", "FrogMIS", "FruitFly"),
                  use.rank=TRUE, group.by=c("Size"), test="quade", correct="finner")

# Data summarization
```

```
res$summary

# Corrected pvalues for the first group
res$corrected.pval[, , 1]

# Grouped data, all vs. control
res <- postHocTest (data=data.blum.2015, control="max", use.rank=FALSE,
                  group.by=c("Size","Radius"), test="wilcoxon", correct="finer")

# Data summarization
res$summary

# Corrected pvalues
res$corrected.pval

# Not grouped data
data(data_gh_2008)
postHocTest (data=data.gh.2008, test="aligned ranks", correct="bergmann")
```

qqplotGaussian

Gaussian distribution quantile-quantile plot

Description

This function creates a quantile-quantile plot to assess the goodness of fit of a Gaussian distribution to a given sample.

Usage

```
qqplotGaussian(data, ...)
```

Arguments

data	List of data points to check
...	The plot is created using <code>ggplot2</code> . This special parameter can be used to pass additional parameters to the <code>geom_point</code> function used to plot the sample points.

Value

A `ggplot` object.

See Also

[plotDensities](#)

Examples

```
## Skewed distribution
sample <- rbeta(100 , 2 , 50)
qqplotGaussian(sample)
## Symmetric distribution
sample <- rbeta(100 , 5 , 5)
qqplotGaussian(sample)
```

quadePost

Quade post hoc raw p-values

Description

This function computes the raw p-values for the post hoc based on Quade's test.

Usage

```
quadePost(data, control = NULL, ...)
```

Arguments

data	Data set (matrix or data.frame) to apply the test. The column names are taken as the groups and the values in the matrix are the samples.
control	Either the number or the name of the column for the control algorithm. If this parameter is not provided, the all vs all comparison is performed.
...	Not used.

Details

The test has been implemented according to the version in Garcia *et al.* (2010), pages 2052,2054

Value

A matrix with all the pairwise raw p-values (all vs. all or all vs. control).

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

Examples

```
data(data_gh_2008)
quadePost(data.gh.2008)
quadePost(data.gh.2008, control=1)
```

quadeTest	<i>Quade's test</i>
-----------	---------------------

Description

This function performs Quade's test for multiple comparisons

Usage

```
quadeTest(data, ...)
```

Arguments

data	Matrix where the test is performed
...	Ignored

Details

The test has been implemented according to the version in Garcia *et al.* (2008).

Value

A list with class "hstest" containing the following components: `statistic`, the value of the statistic used in the test; `parameter`, the two degrees of freedom of the F distribution; `p.value`, the p-value for the test; `method`, a character string indicating what type of test was performed and `data.name`, a character string giving the name of the data.

References

S. Garcia, A. Fernandez, J. Luengo and F. Herrera (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180, 2044-2064.

Examples

```
data(data_gh_2008)
quadeTest(data.gh.2008)
```

rankMatrix	<i>Get the ranking matrix.</i>
------------	--------------------------------

Description

This function returns, given a matrix, the ranking of the columns in each row.

Usage

```
rankMatrix(data, decreasing = TRUE, ...)
```

Arguments

data	The matrix to rank.
decreasing	Logical value indicating whether the top ranked has to be the highest value or not.
...	Not used

Value

A matrix containing the per-row rankings. In case of ties, the mean rank is obtained (e.g, if there is a tie between the 4th and the 5th column, both are assigned a mean rank of 4.5)

Examples

```
data(data_gh_2008)
rankMatrix(data.gh.2008)
```

readComparisonDir	<i>Read data from a directory of comparison-like files</i>
-------------------	------------------------------------------------------------

Description

This function reads the data from all files in a directory. Each file is expected to to be formatted as a comparison file, i.e., the file can have some columns that characterize the problem and one column per algorithm. If each row contain only the result obtained by one algorithm, use the [readExperimentDir](#) function.

Usage

```
readComparisonDir(directory, alg.cols, names, fname.pattern, col.names = NULL,
...)
```


Arguments

directory	Directory where the files are located.
alg.cols	A vector column names or indices indicating which columns contain the results. The rest are assumed as descriptors of the problems
names	List of names for the variables to be extracted from the file name.
fname.pattern	Regular expression to extract information from the file names. It has to be a regular expression that matches the name of the files and where the information to be extracted has to be between brackets. As an example, if the whole file name wants to be used, the expression '([.]*)' can be used. For more example see the examples below or the vignette covering the data loading.
col.names	Vector of names of the columns. If not NULL, the files are assumed not to have a header and the columns are named using this vector.
...	Additional parameters for the read.csv function used to load the data. It can be used, for example, to set the separator (e.g., sep="\t"). Note that the header argument is automatically set according to the col.names argument.

Value

A data.frame where each column represents either a feature of the experiment or the result of running an algorithm. Algorithm columns are placed always at the end of the table.

See Also

[readExperimentFile](#), [readExperimentDir](#), [readComparisonDir](#) and the vignette `vignette(topic="Data_loading_and")`

Examples

```
dir <- paste(system.file("loading_tests", package="scmamp"), "comparison_files", sep="/")
# The format of the files is rgg_size_SIZE_r_RADIUS.out, where variables to extract are in
# capital letters.
list.files(dir)[1]
# The regular expression can be as simple as substituting each variable name in the expression
# above by ([XXX]*), where XXX is the list of symbols that appear in the name.
pattern <- "rgg_size_([0-9]*)_r_(0.[0-9]*)\.out"
var.names <- c("Size", "Radius")
data <- readComparisonDir (directory=dir, alg.cols=1:8, names=var.names,
                          fname.pattern=pattern)

dim(data)
head(data)
```

Description

This function reads the data from a files where two or more algorithms are compared in different problems. The file can have some columns that characterize the problem and one column per algorithm. If each row contain only the result obtained by one algorithm, use the [readExperimentFile](#) function.

Usage

```
readComparisonFile(file, alg.cols, col.names = NULL, ...)
```

Arguments

<code>file</code>	Path of the file to load
<code>alg.cols</code>	A vector column names or indices indicating which columns contain the results. The rest are assumed as descriptors of the problems
<code>col.names</code>	Vector of names of the columns. If not NULL, the files are assumed not to have a header and the columns are named using this vector
<code>...</code>	Additional parameters for the <code>read.csv</code> function used to load the data. It can be used, for example, to set the separator (e.g., <code>sep="\t"</code>). Note that the header argument is automatically set according to the <code>col.names</code> argument.

Value

A data.frame where each column represents either a feature of the experiment or the result of running an algorithm. Algorithm columns are placed always at the end of the table.

See Also

[readExperimentFile](#), [readExperimentDir](#), [readComparisonDir](#) and the vignette `vignette(topic="Data_loading_and`

Examples

```
dir <- system.file("loading_tests", package="scmamp")
file <- paste(dir, "rgg_complete_comparison.out", sep="/")
data <- readComparisonFile(file=file, alg.cols=3:10)
dim(data)
head(data)
```

readExperimentDir	<i>Read data from an experiment-like files in a directory</i>
-------------------	---------------------------------------------------------------

Description

This function reads the data from all the files in a directory. Only one column of results is expected in each file. If the files contain the results of two or more algorithms, see function [readComparisonFile](#). The function can extract information from the file name.

Usage

```
readExperimentDir(directory, names, fname.pattern, alg.var.name, value.col,
  col.names = NULL, ...)
```

Arguments

directory	Directory with the files to load. It should only contain files to load, no other kind of file.
names	List of names for the variables to be extracted from the file name
fname.pattern	Regular expression to extract information from the file names. It has to be a regular expression that matches the name of the files and where the information to be extracted has to be between brackets. As an example, to store the whole file name the expression '([.]*)' can be used. For more example see the examples below or the vignette covering the data loading.
alg.var.name	Name of the variable that defines the algorithm used in the experiment. It can be either one of the variables extracted from the file name or the name of one of the columns in the file.
value.col	Name or index (referred to the column in the file) of the column containing the results.
col.names	Vector of names for the columns. If not provided (or NULL) the names will be read from the first line of the file.
...	Additional parameters for the read.csv function used to load the data. It can be used, for example, to set the separator (e.g., sep="\t"). Note that the header argument is automatically set according to the col.names argument.

Details

Note that all the files should have the same format (same number of columns and, in case they have, same header)

Value

A data.frame where each column represents either a feature of the experiment or the result of running an algorithm. Algorithm columns are placed always at the end of the table.

See Also

[readExperimentFile](#), [readComparisonFile](#), [readComparisonDir](#) and the vignette `vignette(topic="Data_loading_an`

Examples

```
dir <- paste(system.file("loading_tests", package="scmamp"), "experiment_files", sep="/")
# The format of the files is rgg_size_SIZE_r_RADIUS_ALGORITHM.out, where variables
# to extract are in capital letters.
list.files(dir)[1:5]
# The regular expression can be as simple as substituting each variable name in the expression
# above by ([XXX]*), where XXX is the list of symbols that appear in the name.
pattern <- "rgg_size_([0-9]*)_r_(0.[0-9]*)_([a-z,A-Z,1,2]*)\.out"
```

```

var.names <- c("Size", "Radius", "Algorithm")
data <- readExperimentDir (directory=dir, names=var.names, fname.pattern=pattern,
                          alg.var.name="Algorithm", value.col="Evaluation",
                          col.names="Evaluation")

dim(data)
head(data)

```

readExperimentFile *Read data from an experiment-like file*

Description

This function reads the data from a file where each row is an experiment characterized by some variables (one of which should be the algorithm used) and with one and only one numeric result. For files where there is more than one result per line see [readComparisonFile](#).

Usage

```
readExperimentFile(file, alg.col, value.col, col.names = NULL, ...)
```

Arguments

file	Path to the file to read.
alg.col	Name or index of the column corresponding to the algorithm used in the experiment.
value.col	Name or index of the column corresponding to the numerical result of the experiment.
col.names	Vector of names for the columns. If not provided (or NULL) the names will be read from the first line of the file.
...	Additional parameters for the read.csv function used to load the data. It can be used, for example, to set the separator (e.g., sep="\t"). Note that the header argument is automatically set according to the col.names argument.

Value

A data.frame where each column represents either a feature of the experiment or the result of running an algorithm. Algorithm columns are placed always at the end of the table.

See Also

[readExperimentDir](#), [readComparisonFile](#), [readComparisonDir](#) and the vignette `vignette(topic="Data_loading_and")`

Examples

```

dir <- system.file("loading_tests",package="scmamp")
file <- paste(dir , "rgg_complete_experiment.out" , sep="/")
data <- readExperimentFile (file=file, alg.col="Algorithm", value.col="Evaluation")
dim(data)
head(data)

```

 scmamp

Statistical comparison of multiple algorithms

Description

This package has been develop to simplify the statistical assessment of algorithms when tested in different problems. It includes statistical tests, as well as some plotting functions.

Author(s)

Borja Calvo <borja.calvo@ehu.eus>, Guzman Santafe <guzman.santafe@unavarra.es>

Maintainer: Borja Calvo <borja.calvo@ehu.es>

See Also

For an overview of the use see #' vignette(topic= "Statistical_comparison_of_multiple_algorithms_in_multiple_problems") and vignette(topic="Data_loading_and_manipulation",package="scmamp")

 summarizeData

Summarization of data

Description

This is a simple function to apply a summarization function to a matrix or data frame.

Usage

```
summarizeData(data, fun = mean, group.by = NULL, ignore = NULL, ...)
```

Arguments

data	A matrix or data frame to be summarized.
fun	Function to be used in the summarization. It can be any function that, taking as first argument a numeric vector, outputs a numeric value. Typical examples are mean , median , min , max or sd .
group.by	A vector of either column names or column indices according to which the data will be grouped to be summarized.
ignore	A vector of either column names or column indices of the columns that have to be removed from the output.
...	Additional parameters to the summarization function (fun). For example, <code>na.rm=TRUE</code> to indicate that the missing values should be ignored.

Value

A data frame where, for each combination of the values in the columns indicated by `group.by`, each column (except those in `ignore`) contains the summarization of the values in the original matrix that have that combination of values. #' @seealso [filterData](#), [writeTabular](#) and the vignette `vignette(topic="Data_loading_and_manipulation", package="scmamp")`

Examples

```
data(data_blum_2015)
# Group by size and radius. Get the mean and variance of only the last two
# columns.
summarizeData (data.blum.2015, group.by=c("Radius","Size"), ignore=3:8,
              fun=mean, na.rm=TRUE)
summarizeData (data.blum.2015, group.by=c("Radius","Size"), ignore=3:8,
              fun=sd, na.rm=TRUE)
```

 tukeyPost

Tukey post hoc test for ANOVA.

Description

This function computes all the pairwise p-values corrected using Tukey post hoc test.

Usage

```
tukeyPost(data, control = NULL, ...)
```

Arguments

<code>data</code>	Data set (matrix or data.frame) to apply the test. The column names are taken as the groups and the values in the matrix are the samples.
<code>control</code>	Either the number or the name of the column for the control algorithm. If this parameter is not provided, the all vs all comparison is performed.
<code>...</code>	Not used.

Details

The test has been implemented according to Test 22 in Kanji (2006).

Value

A matrix with all the pairwise corrected p-values.

References

G. K. Kanji (2006) *100 Statistical Tests*. SAGE Publications Ltd, 3rd edition.

Examples

```
data(data_gh_2008)
tukeyPost(data.gh.2008)
tukeyPost(data.gh.2008, control=1)
```

tukeyTest	<i>Tukey test</i>
-----------	-------------------

Description

This function performs the Tukey test

Usage

```
tukeyTest(data, alpha = 0.05)
```

Arguments

data	Matrix or data frame where each algorithm is in a column
alpha	Significance level

Details

The test has been implemented according to Test 28 in Kanji (2006).

Value

A list with class "hstest" containing the following components: *statistic*, the value of the statistic used in the test; *method*, a character string indicating what type of test was performed; *data.name*, a character string giving the name of the data and *diff.matrix*, a matrix with all the pairwise absolute difference of average values.

References

Kanji, G. K. (2006) *100 Statistical Tests*. SAGE Publications Ltd, 3rd edition.

Examples

```
data(data_gh_2008)
res <- tukeyTest(data.gh.2008, alpha=0.1)
res
res$diff.matrix
```

wilcoxonSignedTest *Wilcoxon signed-rank test*

Description

This function implements the paired Wilcoxon signed-rank test

Usage

```
wilcoxonSignedTest(x, y, ...)
```

Arguments

x	First sample
y	Second sample
...	Ignored

Details

The test has been implemented according to the version in Demsar (2006), page 7

Value

A list with class "htest" containing the following components: `statistic`, the value of the statistic used in the test; `p.value`, the p-value for the test; `method`, a character string indicating what type of test was performed and `data.name`, a character string giving the name of the data.

References

Demsar, J. (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7, 1-30.

Examples

```
x <- rbeta(50, 2, 20)
y <- x + runif(50) * 0.2
wilcoxonSignedTest(x, y)
```

writeTabular	<i>Write a table in LaTeX format</i>
--------------	--------------------------------------

Description

This is a simple function to create tabular environment in LaTeX

Usage

```
writeTabular(table, file = NULL, format = "g", bold = NULL,
  italic = NULL, mark = NULL, mark.char = "*", na.as = "n/a",
  align = "l", hrule = NULL, vrule = NULL, bty = c("t", "b", "l", "r"),
  print.col.names = TRUE, print.row.names = TRUE, digits = 3,
  wrap.as.table = FALSE, table.position = "h", caption = NULL,
  caption.position = "b", centering = FALSE, label = NULL)
```

Arguments

table	A data frame with the information to write
file	Path of a file. If provided, the tabular is written in the given file. Otherwise, it is written to the standard output
format	Format for the numeric values. The accepted formats are those in the function formatC . The typical values are 'g' to automatically set the format, 'f' for a fixed sized floating point format and 'e' or 'E' for scientific notation
bold	A matrix that matches 'table' in size indicating with TRUE those cells that have to be printed in bold font
italic	A matrix that matches 'table' in size indicating with TRUE those cells that have to be printed in italic
mark	A matrix that matches 'table' in size indicating with TRUE those cells that have to be marked with a superscript symbol
mark.char	Character to be used to mark cells. Note that the superscript is included in a math environment, so this has to be either a character or a valid math command in LaTeX
na.as	Character to be used to write NA values in the table
align	Character indicating the alignment of the columns ('l', 'r' or 'c')
hrule	A vector of positions for the horizontal lines in the tabular. All the lines are drawn after the indicated line. When the column names are included, 0 means drawing a line after the column names. The maximum value is the number of rows - 1 (for a line after the last line see parametr bty)
vrule	Similar to 'hrule' but for vertical lines. . The maximum value is the number of columns - 1 (for a line after the last columns see parametr bty)
bty	Vector indicating which borders should be printed. The vector can contain any of subset of c('l', 'r', 't', 'b'), which represent, respectively, left, right, top and bottom border. If the parameter is set to NULL no border is printed.

<code>print.col.names</code>	Logical value indicating whether the column names have to be printed or not
<code>print.row.names</code>	Logical value indicating whether the row names have to be printed or not
<code>digits</code>	A single number or a numeric vector with the number of digits in each column. Its size has to match the number of the final table, i.e., the columns in 'table' if the row names are not included or the number of columns + 1 if the row names are printed in the final table
<code>wrap.as.table</code>	Logical value indicating whether the latex object has to be wrapped into a table environment
<code>table.position</code>	Character indicating the position of the table ('h': here, 't': top, or 'b': bottom)
<code>caption</code>	Character string containing the caption of the table. If NULL, no caption is printed
<code>caption.position</code>	Character indicating the position of the caption (t: top, the caption is printed over the table; b: bottom, the caption is printed under the table)
<code>centering</code>	Logical value indicating whether the table should be centered in the page
<code>label</code>	Character string containing the label of the table for further references. If NULL, no label is used

Value

LaTeX code to print the table

See Also

[summarizeData](#), [filterData](#) and the vignette `vignette(topic="Data_loading_and_manipulation", package="scamp)`

Examples

```
data(data_blum_2015)
args <- list()
# Write the summarization of the data
args$table <- summarizeData(data.blum.2015, group.by=1:2)

# Set in bold the maximum values per row
bold <- apply(args$table[, -(1:2)], MARGIN=1,
             FUN=function(x) {
               return(x==max(x))
             })
args$bold <- cbind(FALSE, FALSE, t(bold))
# Fixed width, 2 decimals for the values, 0 for the size and 3 for the radius
args$format <- "f"
args$digits <- c(0,3,rep(2, 8))

# Print the column names but not the row names
args$print.row.names <- FALSE
```

```
# Only top and bottom borders
args$bty <- c("t","b")

# Add additional horizontal rules to separate the sizes
args$hrule <- c(0,10,20,30)

# An additional vertical rule to separate size and radius from the results
args$vrule <- 2

# Print the table
do.call(writeTabular, args)
```

Index

adjustBergmannHommel, [3](#), [22](#), [28](#)
adjustFinner, [4](#), [22](#), [28](#)
adjustHolland, [5](#), [22](#), [28](#)
adjustLi, [6](#), [22](#), [28](#)
adjustRom, [7](#), [22](#), [28](#)
adjustShaffer, [8](#), [22](#), [28](#)
anovaTest, [9](#), [22](#)

booleanMatrix, [9](#)

contrastEstimationMatrix, [10](#)
customPost, [11](#)

data.blum.2015, [12](#)
data.gh.2008, [12](#)
data.gh.2010, [13](#)
drawAlgorithmGraph, [13](#), [24–26](#)

exhaustiveSets, [14](#)

filterData, [15](#), [38](#), [42](#)
formatC, [41](#)
friedmanAlignedRanksPost, [16](#), [28](#)
friedmanAlignedRanksTest, [17](#), [22](#)
friedmanPost, [18](#), [28](#)
friedmanTest, [19](#), [22](#)

geom_line, [24](#)
geom_point, [29](#)
ggplot, [24](#), [25](#), [29](#)

imanDavenportTest, [20](#)

max, [37](#)
mean, [37](#)
median, [37](#)
min, [37](#)
multipleComparisonTest, [21](#)

nemenyiTest, [22](#)

plotCD, [14](#), [23](#), [25](#), [26](#)
plotDensities, [24](#), [29](#)
plotPvalues, [14](#), [24](#), [25](#), [26](#)
plotRanking, [14](#), [24](#), [25](#)
postHocTest, [26](#)

qqplotGaussian, [24](#), [29](#)
quadePost, [28](#), [30](#)
quadeTest, [22](#), [31](#)

rankMatrix, [23](#), [32](#)
readComparisonDir, [32](#), [33–36](#)
readComparisonFile, [33](#), [34–36](#)
readExperimentDir, [32–34](#), [34](#), [36](#)
readExperimentFile, [33–35](#), [36](#)

scmamp, [37](#)
scmamp-package (scmamp), [37](#)
sd, [37](#)
summarizeData, [15](#), [37](#), [42](#)

tukeyPost, [28](#), [38](#)
tukeyTest, [39](#)

wilcoxonSignedTest, [40](#)
writeTabular, [15](#), [38](#), [41](#)