Package: PMCMR (via r-universe)

September 9, 2024

Type Package

Title Calculate Pairwise Multiple Comparisons of Mean Rank Sums

Version 4.4

Date 2021-10-02

Description Note, that the 'PMCMR' package is superset by the novel 'PMCMRplus' package. The 'PMCMRplus' package contains all functions from 'PMCMR' and many more parametric and non-parametric multiple comparison procedures, one-factorial trend tests, as well as improved method functions, such as print, summary and plot. The 'PMCMR' package is no longer maintained, but kept for compatibility of reverse depending packages for some time.

Depends R (>= 3.0.0)

Imports stats

Suggests multcompView

Encoding UTF-8

License GPL (>= 3)

NeedsCompilation no

RoxygenNote 7.1.2

Author Thorsten Pohlert [aut, cre]
(<https://orcid.org/0000-0003-3855-3025>)

Maintainer Thorsten Pohlert <thorsten.pohlert@gmx.de>

Repository CRAN

Date/Publication 2021-10-02 15:10:09 UTC

Contents

`get.pvalues` *Get Pvalues from PMCMR Objects*

Description

Returns a vector of pvalues that includes the names of the pairwise groups (i.e. the null hypothesis). The output can be used by `multcompLetters` to find homogeneous groups.

Usage

```
get.pvalues(object, ...)
```

Arguments

`object`	either an object of class `"PMCMR"`, usually, a result of a call to any of the posthoc-tests included in the package PMCMR. Or an object of class `"pairwise.htest"`, a result of a call to `pairwise.prop.test`, `pairwise.t.test` or `pairwise.wilcox.test`.
`...`	further arguments, currently ignored.

Value

a named vector with p-values

`PMCMR-defunct` *PMCMR-defunct*

Description

The functions or methods listed here are no longer part of **PMCMR**. You will find functions and methods in the **PMCMRplus** package `https://cran.r-project.org/package=PMCMRplus`.

Usage

```
dunn.test.control(x, g, p.adjust.method = p.adjust.methods, ...)

jonckheere.test(x, ...)

## Default S3 method:
jonckheere.test(
  x,
  g,
  alternative = c("monotonic", "increasing", "decreasing"),
  ...
)
```

```
posthoc.friedman.conover.test(y, ...)

## Default S3 method:
posthoc.friedman.conover.test(
  y,
  groups,
  blocks,
  p.adjust.method = p.adjust.methods,
  ...
)

posthoc.friedman.nemenyi.test(y, ...)

## Default S3 method:
posthoc.friedman.nemenyi.test(y, groups, blocks, ...)

## S3 method for class 'formula'
posthoc.friedman.nemenyi.test(formula, data, subset, na.action, ...)

durbin.test(y, ...)

## Default S3 method:
durbin.test(y, groups, blocks, ...)

## S3 method for class 'formula'
durbin.test(formula, data, subset, na.action, ...)

posthoc.kruskal.conover.test(x, ...)

## Default S3 method:
posthoc.kruskal.conover.test(x, g, p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
posthoc.kruskal.conover.test(
  formula,
  data,
  subset,
  na.action,
  p.adjust.method = p.adjust.methods,
  ...
)

posthoc.kruskal.dunn.test(x, ...)

## Default S3 method:
posthoc.kruskal.dunn.test(x, g, p.adjust.method = p.adjust.methods, ...)
```

```
## S3 method for class 'formula'
posthoc.kruskal.dunn.test(
  formula,
  data,
  subset,
  na.action,
  p.adjust.method = p.adjust.methods,
  ...
)

posthoc.kruskal.nemenyi.test(x, ...)

## Default S3 method:
posthoc.kruskal.nemenyi.test(x, g, dist = c("Tukey", "Chisquare"), ...)

## S3 method for class 'formula'
posthoc.kruskal.nemenyi.test(
  formula,
  data,
  subset,
  na.action,
  dist = c("Tukey", "Chisquare"),
  ...
)

posthoc.quade.test(y, ...)

## Default S3 method:
posthoc.quade.test(
  y,
  groups,
  blocks,
  dist = c("TDist", "Normal"),
  p.adjust.method = p.adjust.methods,
  ...
)

posthoc.vanWaerden.test(x, ...)

## Default S3 method:
posthoc.vanWaerden.test(x, g, p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
posthoc.vanWaerden.test(
  formula,
  data,
  subset,
  na.action,
```

```
  p.adjust.method = p.adjust.methods,
  ...
)

vanWaerden.test(x, ...)

## Default S3 method:
vanWaerden.test(x, g, ...)

## S3 method for class 'formula'
vanWaerden.test(formula, data, subset, na.action, ...)
```

Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
g	a vector or factor object giving the group for the corresponding elements of x. Ignored if x is a list.
p.adjust.method	Method for adjusting p values (see p.adjust).
...	further arguments to be passed to or from methods.
alternative	The alternative hypothesis.
y	either a numeric vector of data values, or a data matrix.
groups	a vector giving the group for the corresponding elements of y if this is a vector; ignored if y is a matrix. If not a factor object, it is coerced to one.
blocks	a vector giving the block for the corresponding elements of y if this is a vector; ignored if y is a matrix. If not a factor object, it is coerced to one
formula	a formula of the form a ~ b | c, where a, b and c give the data values and corresponding groups and blocks, respectively.
data	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action")´
dist	the test distribution

.Defunct

NA

PMCMR-deprecated *Deprecated Functions in Package* **PMCMR**

Description

These functions are provided for reverse-dependencies issues of other R-packages. They should no longer be used, as actively maintained functions can be found in the package **PMCMRplus**. The functions may be defunct as soon as the next release.

Usage

```
posthoc.durbin.test(y, ...)

## Default S3 method:
posthoc.durbin.test(y, groups, blocks, p.adjust.method = p.adjust.methods, ...)
```

Arguments

y	either a numeric vector of data values, or a data matrix.
...	further arguments to be passed to or from methods.
groups	a vector giving the group for the corresponding elements of y if this is a vector; ignored if y is a matrix. If not a factor object, it is coerced to one.
blocks	a vector giving the block for the corresponding elements of y if this is a vector; ignored if y is a matrix. If not a factor object, it is coerced to one.
p.adjust.method	Method for adjusting p values (see p.adjust).

Value

A list with class "PMCMR"

- method The applied method.
- data.nameThe name of the data.
- p.valueThe two-sided p-value according to the student-t-distribution.
- statisticThe estimated quantiles of the student-t-distribution.
- p.adjust.methodThe applied method for p-value adjustment.

Note

The function does not test, whether it is a true BIBD.

This function does not test for ties.

References

W. J. Conover and R. L. Iman (1979), *On multiple-comparisons procedures*, Tech. Rep. LA-7677-MS, Los Alamos Scientific Laboratory.

W. J. Conover (1999), *Practical nonparametric Statistics*, 3rd. Edition, Wiley.

Examples

```
## Not run:
## Example for an incomplete block design:
## Data from Conover (1999, p. 391).
y <- matrix(c(2, NA, NA, NA, 3, NA, 3, 3,
3, NA, NA, NA,  3, NA, NA,
1,  2, NA, NA, NA,  1,  1, NA,  1,  1,
NA, NA, NA, NA,  2, NA,  2,  1, NA, NA, NA, NA,
3, NA,  2,  1, NA, NA, NA, NA,  3, NA,  2,  2),
ncol=7, nrow=7, byrow=FALSE,
dimnames=list(1:7, LETTERS[1:7]))

posthoc.durbin.test(y)

## End(Not run)
```

`print.PMCMR` *Prints PMCMR objects*

Description

`print` method for class `"PMCMR"`.

Usage

```
## S3 method for class 'PMCMR'
print(x, ...)
```

Arguments

`x`	an object of class `"PMCMR"`, usually, a result of a call to any of the posthoc-tests included in the package PMCMR.
`...`	further arguments, currently ignored.

Value

The function `print.PMCMR` returns the lower triangle of the (adjusted) p-values from any of the posthoc tests included in the package PMCMR.

summary.PMCMR *Summarizing PMCMR objects*

Description

summary method for class "PMCMR".

Usage

```
## S3 method for class 'PMCMR'
summary(object, ...)
```

Arguments

object	an object of class "PMCMR", usually, a result of a call to any of the posthoc-tests included in the package PMCMR.
...	further arguments, currently ignored.

Value

The function summary.PMCMR computes and returns a list of the pairwise comparisons including the H0, the corresponding statistic and the (adjusted) p-value.

Index