

Package ‘PMCMR’

May 19, 2018

Type Package

Title Calculate Pairwise Multiple Comparisons of Mean Rank Sums

Version 4.3

Date 2018-05-19

Author Thorsten Pohlert

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

Description Note, that the 'PMCMR' package is superseded by the novel 'PMCMRplus' package.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 The 'PMCMR' package is no longer maintained, but kept for compatibility of reverse depending packages for some time.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), stats, base

Suggests multcompView, xtable, graphics

Encoding UTF-8

License GPL (>= 3)

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-19 15:20:34 UTC

R topics documented:

PMCMR-package	2
dunn.test.control	3
durbin.test	5
get.pvalues	7
jonckheere.test	8

posthoc.durbin.test	10
posthoc.friedman.conover.test	12
posthoc.friedman.nemenyi.test	14
posthoc.kruskal.conover.test	16
posthoc.kruskal.dunn.test	18
posthoc.kruskal.nemenyi.test	20
posthoc.quade.test	22
posthoc.vanWaerden.test	24
print.PMCMR	26
summary.PMCMR	26
vanWaerden.test	27

Index	29
--------------	-----------

PCMCR-package	<i>Calculate pairwise multiple comparisons between mean rank sums</i>
---------------	-----------------------------------------------------------------------

Description

Note, that the 'PMCMR' package is superseded by the novel 'PMCMRplus' package.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 The 'PMCMR' package is no longer maintained, but kept for compatibility of reverse depending packages for some time.

Details

Package: PMCMR
 Type: Package
 Version: 4.3
 Date: 2018-05-19
 License: >= GPL-3

See the vignette for details.

Author(s)

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

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.
- Janez Demsar (2006), Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, 7, 1-30.

- O.J. Dunn (1964). Multiple comparisons using rank sums. *Technometrics*, 6, 241-252.
- S. A. Glantz (2012), *Primer of Biostatistics*, 7th edition. New York: McGraw Hill.
- N. A. Heckert and J. J. Filliben (2003). NIST Handbook 148: Dataplot Reference Manual, Volume 2: Let Subcommands and Library Functions. National Institute of Standards and Technology Handbook Series, June 2003.
- A. R. Jonckheere (1954). A distribution-free k-sample test against ordered alternatives. *Biometrika*, 41, 133-145.
- P. Nemenyi (1963) *Distribution-free Multiple Comparisons*. Ph.D. thesis, Princeton University.
- D. Quade (1979), Using weighted rankings in the analysis of complete blocks with additive block effects. *Journal of the American Statistical Association*, 74, 680-683.
- Lothar Sachs (1997), *Angewandte Statistik*. Berlin: Springer. Pages: 668-675.

See Also

[posthoc.kruskal.nemenyi.test](#), [posthoc.kruskal.dunn.test](#), [posthoc.kruskal.conover.test](#), [posthoc.friedman.nemenyi.test](#), [posthoc.friedman.conover.test](#), [posthoc.quade.test](#), [dunn.test.control](#), [jonckheere.test](#), [durbin.test](#), [posthoc.durbin.test](#), [kruskal.test](#), [friedman.test](#), [Tukey](#), [TDist](#), [Chisquare](#), [Normal](#), [p.adjust](#), [print.PCMR](#), [summary.PCMR](#), [get.pvalues](#), [multcompLetters](#)

dunn.test.control	<i>Pairwise Test for Multiple Comparisons of Mean Rank Sums with one control (Dunn's-Test)</i>
-------------------	------------------------------------------------------------------------------------------------

Description

Calculate pairwise multiple comparisons with one control according to Dunn.

Usage

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

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. |

Details

For one-factorial designs with samples that do not meet the assumptions for one-way-ANOVA and subsequent post-hoc tests, the Kruskal-Wallis-Test `kruskal.test` can be employed that is also referred to as the Kruskal–Wallis one-way analysis of variance by ranks. Provided that significant differences were detected by this global test, one may be interested in applying post-hoc tests according to Dunn for pairwise multiple comparisons with one control.

See the vignette for details.

Value

A list with class "PMCMR"

<code>method</code>	The applied method.
<code>data.name</code>	The name of the data.
<code>p.value</code>	The two-sided p-value of the standard normal distribution.
<code>statistic</code>	The estimated quantile of the standard normal distribution.
<code>p.adjust.method</code>	The applied method for p-value adjustment.

Note

A tie correction will be employed according to Glantz (2012). As it is the case for multiple testing with one control using `aov`, the user must make sure that the control appears as the first level in the group vector. There is no formula method enclosed.

Author(s)

Thorsten Pohlert

References

O.J. Dunn (1964). Multiple comparisons using rank sums. *Technometrics*, 6, 241-252.

S. A. Glantz (2012), *Primer of Biostatistics*. New York: McGraw Hill.

S. Siegel, N. J. Castellan Jr. (1988), *Nonparametric Statistics for The Behavioral Sciences*. New York: McGraw-Hill.

See Also

[kruskal.test](#), [friedman.test](#), [posthoc.friedman.nemenyi.test](#), [pnorm](#), [p.adjust](#)

Examples

```
##
require(stats)
data(PlantGrowth)
attach(PlantGrowth)
kruskal.test(weight, group)
dunn.test.control(weight, group, "bonferroni")
```

```
detach(PlantGrowth)
rm(PlantGrowth)
```

durbin.test

Durbin test

Description

The omnibus test according to Durbin tests whether k groups (or treatments) in a two-way balanced incomplete block design (BIBD) have identical effects.

Usage

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

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

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

Arguments

<code>y</code>	either a numeric vector of data values, or a data matrix.
<code>groups</code>	a vector giving the group for the corresponding elements of <code>y</code> if this is a vector; ignored if <code>y</code> is a matrix. If not a factor object, it is coerced to one.
<code>blocks</code>	a vector giving the block for the corresponding elements of <code>y</code> if this is a vector; ignored if <code>y</code> is a matrix. If not a factor object, it is coerced to one.
<code>formula</code>	a formula of the form $a \sim b \mid c$, where <code>a</code> , <code>b</code> and <code>c</code> give the data values and corresponding groups and blocks, respectively.
<code>data</code>	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula <code>formula</code> . By default the variables are taken from <code>environment(formula)</code> .
<code>subset</code>	an optional vector specifying a subset of observations to be used.
<code>na.action</code>	a function which indicates what should happen when the data contain NAs. Defaults to <code>getOption("na.action")</code> .
<code>...</code>	further arguments to be passed to or from methods.

Details

The `friedman.test` can be used to test k groups (treatments) for identical effects in a two-way balanced complete block design. In the case of an two-way balanced incomplete block design, the Durbin test can be employed. The H_0 is rejected, if at least one group (treatment) is significantly different. The Durbin test is equivalent to the Friedman test in the case of a two-way balanced complete block design.

If y is a matrix, than the columns refer to the groups (treatment) and the rows indicate the block.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR":

<code>method</code>	The applied method.
<code>data.name</code>	The name of the data.
<code>p.value</code>	The p-value according to the studentized range distribution.
<code>statistic</code>	The estimated upper quantile of the studentized range distribution.
<code>p.adjust.method</code>	Defaults to "none"

Note

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

This function does not test for ties.

Author(s)

Thorsten Pohlert

References

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

N. A. Heckert and J. J. Filliben (2003). NIST Handbook 148: Dataplot Reference Manual, Volume 2: Let Subcommands and Library Functions. National Institute of Standards and Technology Handbook Series, June 2003.

See Also

[friedman.test](#), [posthoc.durbin.test](#)

Examples

```
## 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]))
y
durbin.test(y)

## Example for a complete block design:
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
##
y <- matrix(c(
3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
26.65),nrow=6, ncol=6,
dimnames=list(1:6,LETTERS[1:6]))
print(y)
friedman.test(y)
durbin.test(y)

```

get.pvalues

Prints 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

out	a named vector with p-values
-----	------------------------------

See Also

[multcompLetters](#), [xtable](#), [pairwise.prop.test](#), [pairwise.t.test](#), [pairwise.wilcox.test](#)

Examples

```
data(InsectSprays)
attach(InsectSprays)
out <- posthoc.kruskal.dunn.test(count ~ spray, p.adjust="bonf")
out.p <- get.pvalues(out)
out.p

### a barplot, significant level at p < 0.05
require(multcompView)
out.mcV <- multcompLetters(out.p, threshold=0.05)
Rij <- rank(count)
Rj.mean <- tapply(Rij, spray, mean)
ti <- paste(out$method, "\nP-adjustment method:", out$p.adjust.method)
xx <- barplot(Rj.mean, ylim=c(0, 1.2* max(Rj.mean)),
             xlab="Spray", ylab="Mean rank", main=ti)
yy <- Rj.mean + 3
text(xx, yy, lab=out.mcV$Letters)

## table format
dat <- data.frame(Group = names(Rj.mean),
                  meanRj = Rj.mean,
                  M = out.mcV$Letters)

dat

## LaTeX table
require(xtable)
xtable(dat, caption=ti, digits=1)

detach(InsectSprays)
```

jonckheere.test	<i>Jonckheere-Terpstrata test for testing k independent samples against ordered alternatives</i>
-----------------	--------------------------------------------------------------------------------------------------

Description

Applies the Jonckheere-Terpstrata test.

Usage

```
jonckheere.test(x, ...)

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


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.
alternative	The alternative hypothesis. Defaults to monotonic.
...	further arguments to be passed to or from methods.

Details

The global H0, the k medians (med) are equal, is tested against the HA: $\text{med}_1 \leq \text{med}_2 \leq \dots \leq \text{med}_k$ (increasing trend; one-sided test). It can also be tested for decreasing trend (one-sided test) and monotonic trend (two sided test).

See the vignette for details.

Value

A list with class "PMCMR"

method	The applied method.
data.name	The name of the data.
p.value	The p-value of the standard normal distribution.
statistic	The estimated quantile of the standard normal distribution.
alternative	The alternative hypothesis.

Author(s)

Thorsten Pohlert

References

- A. R. Jonckheere (1954). A distribution-free k-sample test against ordered alternatives. *Biometrika*, 41, 133-145.
- L. Sachs (1997), *Angewandte Statistik*. Berlin: Springer.

See Also

[kruskal.test](#)

Examples

```
##
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145, 110, 125,
      143, 148, 151, 136, 139, 149, 160,
      174)
g <- as.factor(c(rep(1,5), rep(2,5), rep(3,5)))
levels(g) <- c("A", "B", "C")
jonckheere.test(x , g, "increasing")
```

posthoc.durbin.test *Posthoc Durbin test*

Description

Pairwise post-hoc test for multiple comparisons of rank sums according to Durbin and Conover for a two-way balanced incomplete block design (BIBD).

Usage

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

Arguments

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

Details

In the case of an two-way balanced incomplete block design, the Durbin test, `durbin.test` can be employed. The H_0 is rejected, if at least one group (treatment) is significantly different. The pairwise multiple comparisons are conducted with this function. The `posthoc.durbin.test` is equivalent to the `posthoc.friedman.conover.test` in the case of a two-way balanced complete block design.

If `y` is a matrix, than the columns refer to the groups (treatment) and the rows indicate the block.

The statistics refer to the student-t-distribution (TDist).

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

method	The applied method.
data.name	The name of the data.
p.value	The two-sided p-value according to the student-t-distribution.
statistic	The estimated quantiles of the student-t-distribution.
p.adjust.method	The 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.

Author(s)

Thorsten Pohlert

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.

See Also

[durbin.test](#), [friedman.test](#), [posthoc.friedman.nemenyi.test](#), [posthoc.friedman.conover.test](#), [TDist](#) [p.adjust](#)

Examples

```
## 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]))
y
durbin.test(y)
posthoc.durbin.test(y, p.adj="none")

## Example for a complete block design:
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics (A to F, treatment).
## The responses are the Na-concentration (mval)
```

```
## in the urine measured 2 hours after each treatment.
##
y <- matrix(c(
  3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
  23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
  26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
  32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
  26.65),nrow=6, ncol=6,
dimnames=list(1:6,LETTERS[1:6]))
print(y)
friedman.test(y)
durbin.test(y)
posthoc.durbin.test(y, p.adj="none")
posthoc.friedman.conover.test(y, p.adj="none")
```

```
posthoc.friedman.conover.test
```

Pairwise post-hoc Test for Multiple Comparisons of Rank Sums for Unreplicated Blocked Data (Conover-Test)

Description

Calculate pairwise comparisons using Conover post-hoc test for unreplicated blocked data. This test is usually conducted post-hoc after significant results of the `friedman.test`. The statistics refer to the student-t-distribution (TDist).

Usage

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

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

Arguments

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

Details

A one-way ANOVA with repeated measures that is also referred to as ANOVA with unreplicated block design can also be conducted via the `friedman.test`. The consequent post-hoc pairwise multiple comparison test according to Conover is conducted with this function.

If `y` is a matrix, than the columns refer to the treatment and the rows indicate the block.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

<code>method</code>	The applied method.
<code>data.name</code>	The name of the data.
<code>p.value</code>	The two-sided p-value according to the student-t-distribution.
<code>statistic</code>	The estimated quantiles of the student-t-distribution.
<code>p.adjust.method</code>	The applied method for p-value adjustment.

Note

This function does not test for ties.

Author(s)

Thorsten Pohlert

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.

See Also

[friedman.test](#), [posthoc.friedman.nemenyi.test](#), [TDist](#) [p.adjust](#)

Examples

```
##
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics
## (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
##
y <- matrix(c(
  3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
  23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
```

```

26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
26.65),nrow=6, ncol=6,
dimnames=list(1:6,c("A","B","C","D","E","F")))
print(y)
friedman.test(y)
posthoc.friedman.conover.test(y=y, p.adjust="none")

```

```
posthoc.friedman.nemenyi.test
```

*Pairwise post-hoc Test for Multiple Comparisons of Mean Rank Sums
for Unreplicated Blocked Data (Nemenyi-Test)*

Description

Calculate pairwise comparisons using Nemenyi post-hoc test for unreplicated blocked data. This test is usually conducted post-hoc after significant results of the `friedman.test`. The statistics refer to upper quantiles of the studentized range distribution (Tukey).

Usage

```

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, ...)

```

Arguments

<code>y</code>	either a numeric vector of data values, or a data matrix.
<code>groups</code>	a vector giving the group for the corresponding elements of <code>y</code> if this is a vector; ignored if <code>y</code> is a matrix. If not a factor object, it is coerced to one.
<code>blocks</code>	a vector giving the block for the corresponding elements of <code>y</code> if this is a vector; ignored if <code>y</code> is a matrix. If not a factor object, it is coerced to one.
<code>formula</code>	a formula of the form <code>a ~ b c</code> , where <code>a</code> , <code>b</code> and <code>c</code> give the data values and corresponding groups and blocks, respectively.
<code>data</code>	an optional matrix or data frame (or similar: see <code>model.frame</code>) containing the variables in the formula <code>formula</code> . By default the variables are taken from <code>environment(formula)</code> .
<code>subset</code>	an optional vector specifying a subset of observations to be used.
<code>na.action</code>	a function which indicates what should happen when the data contain NAs. Defaults to <code>getOption("na.action")</code> .
<code>...</code>	further arguments to be passed to or from methods.

Details

A one-way ANOVA with repeated measures that is also referred to as ANOVA with unreplicated block design can also be conducted via the `friedman.test`. The consequent post-hoc pairwise multiple comparison test according to Nemenyi is conducted with this function.

If y is a matrix, than the columns refer to the treatment and the rows indicate the block.

See `vignette("PMCMR")` for details.

Let R_j and n_j denote the sum of Friedman-ranks and the sample size of the j -th group, respectively, then a difference between two groups is significant on the level of α , if the following inequality is met:

$$\left| \frac{R_i}{n_i} - \frac{R_j}{n_j} \right| > \frac{q_{\infty; k; \alpha}}{\sqrt{2}} \sqrt{\frac{k(k+1)}{6n}}$$

with k the number of groups (or treatments) and n the total number of data.

Value

A list with class "PMCMR"

<code>method</code>	The applied method.
<code>data.name</code>	The name of the data.
<code>p.value</code>	The p-value according to the studentized range distribution.
<code>statistic</code>	The estimated upper quantile of the studentized range distribution.
<code>p.adjust.method</code>	Defaults to "none"

Note

This function does not test for ties.

Author(s)

Thorsten Pohlert

References

Janez Demsar (2006), Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, 7, 1-30.

P. Nemenyi (1963) *Distribution-free Multiple Comparisons*. Ph.D. thesis, Princeton University.

Lothar Sachs (1997), *Angewandte Statistik*. Berlin: Springer. Pages: 668-675.

See Also

[friedman.test](#), [kruskal.test](#), [posthoc.kruskal.nemenyi.test](#), [Tukey](#)

Examples

```
##
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
##
y <- matrix(c(
  3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
  23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
  26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
  32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
  26.65),nrow=6, ncol=6,
  dimnames=list(1:6,c("A","B","C","D","E","F")))
print(y)
friedman.test(y)
posthoc.friedman.nemenyi.test(y)
```

```
posthoc.kruskal.conover.test
```

*Pairwise Test for Multiple Comparisons of Mean Rank Sums
(Conover's-Test)*

Description

Calculate pairwise multiple comparisons between group levels according to Conover.

Usage

```
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, ...)
```

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.
formula	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.

data	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula <code>formula</code> . By default the variables are taken from <code>environment(formula)</code> .
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 <code>getOption("na.action")</code> .
p.adjust.method	Method for adjusting p values (see p.adjust).
...	further arguments to be passed to or from methods.

Details

For one-factorial designs with samples that do not meet the assumptions for one-way-ANOVA and subsequent post-hoc tests, the Kruskal-Wallis-Test [kruskal.test](#) can be employed that is also referred to as the Kruskal–Wallis one-way analysis of variance by ranks. Provided that significant differences were detected by this global test, one may be interested in applying post-hoc tests according to Conover for pairwise multiple comparisons of the ranked data.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

method	The applied method.
data.name	The name of the data.
p.value	The two-sided p-value of the student-t-distribution.
statistic	The estimated quantile of the student-t-distribution.
p.adjust.method	The applied method for p-value adjustment.

Note

A tie correction will be employed according to Conover (1979).

Author(s)

Thorsten Pohlert

References

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

See Also

[kruskal.test](#), [posthoc.kruskal.nemenyi.test](#), [posthoc.kruskal.dunn.test](#), [TDist](#), [p.adjust](#)

Examples

```
##
require(stats)
data(InsectSprays)
attach(InsectSprays)
kruskal.test(count, spray)
posthoc.kruskal.conover.test(count, spray, "bonferroni")
detach(InsectSprays)
rm(InsectSprays)
## Formula Interface
posthoc.kruskal.conover.test(count ~ spray, data = InsectSprays, p.adjust="bonf")
```

```
posthoc.kruskal.dunn.test
```

Pairwise Test for Multiple Comparisons of Mean Rank Sums (Dunn's-Test)

Description

Calculate pairwise multiple comparisons between group levels according to Dunn.

Usage

```
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, ...)
```

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.
formula	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
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 <code>getOption("na.action")</code> .
p.adjust.method	Method for adjusting p values (see p.adjust).
...	further arguments to be passed to or from methods.

Details

For one-factorial designs with samples that do not meet the assumptions for one-way-ANOVA and subsequent post-hoc tests, the Kruskal-Wallis-Test [kruskal.test](#) can be employed that is also referred to as the Kruskal–Wallis one-way analysis of variance by ranks. Provided that significant differences were detected by this global test, one may be interested in applying post-hoc tests according to Dunn for pairwise multiple comparisons of the ranked data.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

method	The applied method.
data.name	The name of the data.
p.value	The two-sided p-value of the standard normal distribution.
statistic	The estimated quantile of the standard normal distribution.
p.adjust.method	The applied method for p-value adjustment.

Note

A tie correction will be employed according to Glantz (2012).

Author(s)

Thorsten Pohlert

References

- O.J. Dunn (1964). Multiple comparisons using rank sums. *Technometrics*, 6, 241-252.
- S. A. Glantz (2012), *Primer of Biostatistics*. New York: McGraw Hill.

See Also

[kruskal.test](#), [friedman.test](#), [posthoc.friedman.nemenyi.test](#), [pnorm](#), [p.adjust](#)

Examples

```
##
require(stats)
data(InsectSprays)
attach(InsectSprays)
kruskal.test(count, spray)
posthoc.kruskal.dunn.test(count, spray, "bonferroni")
detach(InsectSprays)
rm(InsectSprays)
## Formula Interface
posthoc.kruskal.dunn.test(count ~ spray, data = InsectSprays, p.adjust="bonf")
```

```
posthoc.kruskal.nemenyi.test
```

*Pairwise Test for Multiple Comparisons of Mean Rank Sums
(Nemenyi-Tests)*

Description

Calculate pairwise multiple comparisons between group levels. These tests are sometimes referred to as Nemenyi-tests for multiple comparisons of (mean) rank sums of independent samples.

Usage

```
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"), ...)
```

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.
formula	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
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 <code>getOption("na.action")</code> .
...	further arguments to be passed to or from methods.
dist	the method for determining the p-value. The default distribution is "Tukey", else "Chisq".

Details

For one-factorial designs with samples that do not meet the assumptions for one-way-ANOVA and subsequent post-hoc tests, the Kruskal-Wallis-Test `kruskal.test` can be employed that is also referred to as the Kruskal–Wallis one-way analysis of variance by ranks. Provided that significant differences were detected by this global test, one may be interested in applying post-hoc tests according to Nemenyi for pairwise multiple comparisons of the ranked data.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

method	The applied method.
data.name	The name of the data.
p.value	The p-value according to the studentized range distribution.
statistic	The estimated upper quantile of the studentized range distribution. (or quantile of Chisq distribution)
p.adjust.method	Defaults to "none"

Note

Only for `method = "Chisq"` a tie correction is employed.

Author(s)

Thorsten Pohlert

References

Lothar Sachs (1997), *Angewandte Statistik*. Berlin: Springer. Pages: 395-397, 662-664.

See Also

[kruskal.test](#), [friedman.test](#), [posthoc.friedman.nemenyi.test](#), Tukey, [Chisquare](#)

Examples

```
##
require(stats)
data(InsectSprays)
attach(InsectSprays)
kruskal.test(count, spray)
posthoc.kruskal.nemenyi.test(count, spray)
posthoc.kruskal.nemenyi.test(count, spray, "Chisq")
detach(InsectSprays)
rm(InsectSprays)
## Formula Interface
posthoc.kruskal.nemenyi.test(count ~ spray, data = InsectSprays, dist="Tukey")
```

posthoc.quade.test	<i>Pairwise post-hoc Test for Multiple Comparisons of Rank Sums for Unreplicated Blocked Data (Quade multiple comparison test)</i>
--------------------	------------------------------------------------------------------------------------------------------------------------------------

Description

Calculate pairwise comparisons using Quade's post-hoc test for unreplicated blocked data. This test is usually conducted post-hoc after significant results of the omnibus test, `quade.test`.

Usage

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

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

Arguments

<code>y</code>	either a numeric vector of data values, or a data matrix.
<code>groups</code>	a vector giving the group for the corresponding elements of <code>y</code> if this is a vector; ignored if <code>y</code> is a matrix. If not a factor object, it is coerced to one.
<code>blocks</code>	a vector giving the block for the corresponding elements of <code>y</code> if this is a vector; ignored if <code>y</code> is a matrix. If not a factor object, it is coerced to one.
<code>dist</code>	the method for determining the p-value. The default distribution is "TDist", else "Normal".
<code>p.adjust.method</code>	Method for adjusting p values (see p.adjust).
<code>...</code>	further arguments to be passed to or from methods.

Details

A one-way ANOVA with repeated measures that is also referred to as ANOVA with unreplicated block design can also be conducted via the `quade.test`. The consequent post-hoc pairwise multiple comparison test according to Quade is conducted with this function.

If `y` is a matrix, than the columns refer to the treatment and the rows indicate the block.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

<code>method</code>	The applied method.
<code>data.name</code>	The name of the data.
<code>p.value</code>	The two-sided p-value according to the selected distribution.
<code>statistic</code>	The estimated quantiles of the selected distribution.
<code>p.adjust.method</code>	The applied method for p-value adjustment.

Author(s)

Thorsten Pohlert

References

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

N. A. Heckert and J. J. Filliben (2003). NIST Handbook 148: Dataplot Reference Manual, Volume 2: Let Subcommands and Library Functions. National Institute of Standards and Technology Handbook Series, June 2003.

D. Quade (1979), Using weighted rankings in the analysis of complete blocks with additive block effects. *Journal of the American Statistical Association*, 74, 680-683.

See Also

[quade.test](#), [Normal](#), [TDist](#) [p.adjust](#)

Examples

```
## Conover (1999, p. 375f):
## Numbers of five brands of a new hand lotion sold in seven stores
## during one week.
y <- matrix(c( 5,  4,  7, 10, 12,
              1,  3,  1,  0,  2,
              16, 12, 22, 22, 35,
              5,  4,  3,  5,  4,
              10,  9,  7, 13, 10,
              19, 18, 28, 37, 58,
              10,  7,  6,  8,  7),
            nrow = 7, byrow = TRUE,
```

```

      dimnames =
      list(Store = as.character(1:7),
           Brand = LETTERS[1:5]))
y
quade.test(y)
posthoc.quade.test(y, dist="TDist", p.adj="none")

```

```
posthoc.vanWaerden.test
```

Pairwise Test for Multiple Comparisons of normal scores (van der Waerden test)

Description

Calculate pairwise multiple comparisons between group levels according to van der Waerden.

Usage

```

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, ...)

```

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.
formula	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
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 <code>environment(formula)</code> .
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 <code>getOption("na.action")</code> .
p.adjust.method	Method for adjusting p values (see p.adjust).
...	further arguments to be passed to or from methods.

Details

For one-factorial designs with samples that do not meet the assumptions for one-way-ANOVA and subsequent post-hoc tests, the van der Waerden test `vanWaerden.test` using normal scores can be employed. Provided that significant differences were detected by this global test, one may be interested in applying post-hoc tests according to van der Waerden for pairwise multiple comparisons of the group levels.

First, the data are ranked according to Kruskal-Wallis. Second, the ranks are transformed to normal scores. The group means of normal scores and the total variance is used to calculate quantiles of the student-t-distribution and consequent p-values.

See `vignette("PMCMR")` for details.

Value

A list with class "PMCMR"

<code>method</code>	The applied method.
<code>data.name</code>	The name of the data.
<code>p.value</code>	The two-sided p-value of the student-t-distribution.
<code>statistic</code>	The estimated quantile of the student-t-distribution.
<code>p.adjust.method</code>	The applied method for p-value adjustment.

Note

There is no tie correction applied in this function.

Author(s)

Thorsten Pohlert

References

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

See Also

`kruskal.test`, `vanWaerden.test`, `posthoc.kruskal.nemenyi.test`, `posthoc.kruskal.dunn.test`, `TDist`, `p.adjust`

Examples

```
##
require(stats)
data(InsectSprays)
attach(InsectSprays)
vanWaerden.test(count, spray)
posthoc.vanWaerden.test(count, spray, "bonferroni")
```

```
detach(InsectSprays)
rm(InsectSprays)
```

```
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.

Examples

```
out <- posthoc.kruskal.conover.test(count ~ spray,
data = InsectSprays, p.adjust="bonf")
print(out)
```

```
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 H_0 , the corresponding statistic and the (adjusted) p-value.

Examples

```
out <- posthoc.kruskal.conover.test(count ~ spray,
data = InsectSprays, p.adjust="bonf")
summary(out)
```

vanWaerden.test *van der Waerden's normal scores test*

Description

Performs a van der Waerden's normal scores test.

Usage

```
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.
 formula a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
 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")`.
 ... further arguments to be passed to or from methods.

Details

vanWaerden.test performs a van der Waerden test of the null that the location parameters of the distribution of x are the same in each group (sample). The alternative is that they differ in at least one.

See vignette("PMCMR") for details.

Value

A list with class "hstest" that contains the following components:

statistic	the van der Waerden chi-squared statistic.
parameter	the degrees of freedom of the approximate chi-squared distribution of the test statistic.
p.value	the p-value of the test.
method	the character string "Van der Waerden normal scores test".
data.name	a character string giving the names of the data.

Note

A tie correction is not applied in this function.

Author(s)

Thorsten Pohlert

References

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

See Also

[kruskal.test](#)

Examples

```
##
require(stats)
data(InsectSprays)
attach(InsectSprays)
vanWaerden.test(count, spray)
posthoc.vanWaerden.test(count, spray, "bonferroni")
detach(InsectSprays)
rm(InsectSprays)
```

Index

*Topic **h**test

- dunn.test.control, 3
- durbin.test, 5
- jonckheere.test, 8
- PCMCr-package, 2
- posthoc.durbin.test, 10
- posthoc.friedman.conover.test, 12
- posthoc.friedman.nemenyi.test, 14
- posthoc.kruskal.conover.test, 16
- posthoc.kruskal.dunn.test, 18
- posthoc.kruskal.nemenyi.test, 20
- posthoc.quade.test, 22
- posthoc.vanWaerden.test, 24
- vanWaerden.test, 27

*Topic **m**ethods

- summary.PCMCR, 26

*Topic **n**onparametric

- dunn.test.control, 3
- durbin.test, 5
- jonckheere.test, 8
- PCMCr-package, 2
- posthoc.durbin.test, 10
- posthoc.friedman.conover.test, 12
- posthoc.friedman.nemenyi.test, 14
- posthoc.kruskal.conover.test, 16
- posthoc.kruskal.dunn.test, 18
- posthoc.kruskal.nemenyi.test, 20
- posthoc.quade.test, 22
- posthoc.vanWaerden.test, 24
- vanWaerden.test, 27

*Topic **p**ackage

- PCMCr-package, 2

*Topic **p**rint

- print.PCMCR, 26
- summary.PCMCR, 26

*Topic **u**tilities

- get.pvalues, 7

Chisquare, 3, 21

dunn.test.control, 3, 3

durbin.test, 3, 5, 11

friedman.test, 3, 4, 6, 11, 13, 15, 19, 21

get.pvalues, 3, 7

jonckheere.test, 3, 8

kruskal.test, 3, 4, 9, 15, 17, 19, 21, 25, 28

model.frame, 5, 14, 17, 18, 20, 24, 27

multcompLetters, 3, 7, 8

Normal, 3, 23

p.adjust, 3, 4, 10–13, 17, 19, 22–25

pairwise.prop.test, 7, 8

pairwise.t.test, 7, 8

pairwise.wilcox.test, 7, 8

PCMCr (PCMCr-package), 2

PCMCr-package, 2

pnorm, 4, 19

posthoc.durbin.test, 3, 6, 10

posthoc.friedman.conover.test, 3, 11, 12

posthoc.friedman.nemenyi.test, 3, 4, 11, 13, 14, 19, 21

posthoc.kruskal.conover.test, 3, 16

posthoc.kruskal.dunn.test, 3, 17, 18, 25

posthoc.kruskal.nemenyi.test, 3, 15, 17, 20, 25

posthoc.quade.test, 3, 22

posthoc.vanWaerden.test, 24

print.PCMCR, 3, 26

quade.test, 23

summary.PCMCR, 3, 26

TDist, 3, 11, 13, 17, 23, 25

Tukey, 3, 15, 21

vanWaerden.test, [25](#), [27](#)

xtable, [8](#)