Package ‘PMCMR’

January 18, 2018

Type Package
Title Calculate Pairwise Multiple Comparisons of Mean Rank Sums
Version 4.2
Date 2018-01-15
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Description Note, that the ‘PMCMR’ package is superseded by the novel ‘PMCMRplus’ package.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 The ‘PMCMR’ package is no longer maintained, but kept for compatibility of reverse depending packages for some time.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 2018-01-18 00:05:17 UTC

R topics documented:

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

type: Package

version: 4.1

date: 2016-01-06

license: >= GPL-3

See the vignette for details.

Author(s)

Thorsten Pohlert

Maintainer: <thorsten.pohlert@gmx.de>
References


See Also


---

**dunn.test.control**

Pairwise Test for Multiple Comparisons of Mean Rank Sums with one control (Dunn’s-Test)

**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.
- `...`: 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"

- `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.

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


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

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

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. De-
                     faults to getOption("na.action").
...                  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 H0 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":

  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.
  p.adjust.method      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


See Also

friedman.test, posthoc.durbin.test
get.pvalues

Examples

## Example for an incomplete block design:
## Data from Conover (1999, p. 391).
y <- matrix(c(2,NA,NA,NA,3, NA, 3, 3, NA, NA, NA, 3, NA, NA, 1, 2, NA, NA, NA, 1, 1, NA, 1, 1, 
), 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)

g et. 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.wilcoxon.test.
...

further arguments, currently ignored.
jonckheere.test

Jonckheere-Terpstra test for testing \( k \) independent samples against ordered alternatives

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$method, "\n\n\nadjustment method: ", out$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)
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 $H_0$, the k medians (med) are equal, is tested against the $H_A$: med_1 \leq med_2 \leq \ldots \leq 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


See Also

kruskal.test
Examples

```r
## 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**

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

**Arguments**

- `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.

- `p.adjust.method`  
Method for adjusting p values (see `p.adjust`).

- `...`  
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 H0 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

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,NA,NA,3,NA,NA,1,2,NA,NA,NA,1,1,NA,1,1,NA,NA,NA,2,NA,2,1,NA,NA,NA,NA,3,NA,2,1,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.

```r
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(LETTERS[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

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

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

### Arguments

- **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.

p.adjust.method
               Method for adjusting p values (see p.adjust).

...                 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"

  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

This function does not test for ties.

Author(s)

Thorsten Pohlert

References


See Also

friedman.test, posthoc.friedman.nemenyi.test, TDist p.adjust
Examples

```r
## Sachs, 1997, p. 675
## Six persons (blockI 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, 28.86, 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, 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

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

#### Default S3 method:

```r
posthoc.friedman.nemenyi.test(y, groups, blocks, ...)
```

#### S3 method for class 'formula'

```r
posthoc.friedman.nemenyi.test(formula, data, subset, na.action, ...)
```

### Arguments

- **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
formula
data
subset
na.action
...
References


See Also

*friedman.test, kruskal.test, posthoc.kruskal.nemenyi.test, Tukey*

Examples

```r
## 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

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

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

## S3 method for class 'formula'

posthoc.kruskal.conover.test

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 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. De-
defaults to getOption("na.action").
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 re-
ferred 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 ac-
cording 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).
posthoc.kruskal.dunn.test

Author(s)
Thorsten Pohlert

References

See Also
kruskal.test, posthoc.kruskal.nemenyi.test, posthoc.kruskal.dunn.test, TDist.p.adjust

Examples
```r
##
## 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 getOption("na.action").
- **p.adjust.method**: Method for adjusting p values (see p.adjust).

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


See Also

`kruskal.test`, `friedman.test`, `posthoc.friedman.nemenyi.test`, `pnorm`, `p.adjust`

Examples

```r
##
require(stats)
data(InsectSprays)
attach(InsectSprays)
krukasal.test(count, spray)

posthoc.kruskaal.dunn.test(count, spray, "bonferroni")
detach(InsectSprays)

## Formula Interface

posthoc.kruskaal.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

```r
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 `NA`s. Defaults to `getOption("na.action")`.
- **...**: 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


See Also

kruskal.test, friedman.test, posthoc.friedman.nemenyi.test, Tukey, Chisquare

Examples

```r
## 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**  
Pairwise post-hoc Test for Multiple Comparisons of Rank Sums for Unreplicated Blocked Data (Quade multiple comparison test)

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

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

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

Arguments

- **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.
- **dist**: the method for determining the p-value. The default distribution is "TDist", else "Normal".
posthoc.quade.test

p.adjust.method
   Method for adjusting p values (see \texttt{p.adjust}).

\ldots
   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 \texttt{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"

\begin{itemize}
   \item \texttt{method} The applied method.
   \item \texttt{data.name} The name of the data.
   \item \texttt{p.value} The two-sided p-value according to the selected distribution.
   \item \texttt{statistic} The estimated quantiles of the selected distribution.
   \item \texttt{p.adjust.method} The applied method for p-value adjustment.
\end{itemize}

Author(s)

Thorsten Pohlert

References


See Also

\texttt{quade.test, Normal, TDist p.adjust}

Examples

\begin{verbatim}
   ## 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),
                 nrow=7, byrow=TRUE)
   quade.test(y)
\end{verbatim}
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
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. De-
faults togetOption("na.action").
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"

- `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**

There is no tie correction applied in this function.

**Author(s)**

Thorsten Pohlert

**References**


**See Also**

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

**Description**

print method for class "PMCMR".

**Usage**

```r
## 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**

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

**Summary PMCMR objects**

**Description**

Summary method for class "PMCMR".

**Usage**

```r
## 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.

**Examples**

```r
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**

```r
vanWaerden.test(x, ...)
```

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

```r
## 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 "htest" 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

See Also

kruskal.test

Examples

```r
data(insectSprays)
attach(insectSprays)
vanWaerden.test(count, spray)
posthoc.vanWaerden.test(count, spray, "bonferroni")
detach(insectSprays)
rm(insectSprays)
```
Index

*Topic htest
  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 methods
  summary.PMCMR, 27

*Topic nonparametric
  dunn.test.control, 3
durbin.test, 5
dunn.test.control, 3
friedman.test, 3, 5, 6, 11, 13, 16, 20, 22
get.pvalues, 3
jonckheere.test, 3, 8
kruskal.test, 3–5, 9, 16–22, 25, 29
model.frame, 5, 15, 17, 19, 21, 24, 28
multcompLetters, 3, 7, 8
Normal, 3, 23

p.adjust, 3–5, 10, 11, 13, 17–20, 23, 25
pairwise.prop.test, 7, 8
pairwise.t.test, 7, 8
pairwise.wilcox.test, 7, 8
PCMCR (PCMCR-package), 2
PCMCR-package, 2
pnorm, 5, 20
posthoc.durbin.test, 3, 6, 10
posthoc.friedman.conover.test, 3, 11, 12
posthoc.friedman.nemenyi.test, 3, 5, 11, 13, 14, 20, 22
posthoc.kruskal.conover.test, 3, 16
posthoc.kruskal.dunn.test, 3, 18, 18, 25
posthoc.kruskal.nemenyi.test, 3, 16, 18, 18, 20, 25
posthoc.quade.test, 3, 22
posthoc.vanWaerden.test, 24
print.PMCMR, 3, 26
quade.test, 23
summary.PMCMR, 3, 27

TDist, 3, 11, 13, 18, 23, 25
Tukey, 3, 16, 22

Chisquare, 3, 22
INDEX

vanWaverden.test, 25, 27

xtable, 8