

# Package 'PMCMRplus'

January 17, 2018

**Type** Package

**Title** Calculate Pairwise Multiple Comparisons of Mean Rank Sums  
Extended

**Version** 1.0.0

**Date** 2018-01-15

**Author** Thorsten Pohlert

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

**Description** For one-way layout experiments the one-way ANOVA can be performed as an omnibus test. All-pairs multiple comparisons tests (Tukey-Kramer test, Scheffe test, LSD-test) and many-to-one tests (Dunnett test) for normally distributed residuals and equal within variance are available. Furthermore, all-pairs tests (Games-Howell test, Tamhane's T2 test, Dunnett T3 test, Ury-Wiggins-Hochberg test) and many-to-one (Tamhane-Dunnett Test) for normally distributed residuals and heterogeneous variances are provided. Van der Waerden's normal scores test for omnibus, all-pairs and many-to-one tests is provided for non-normally distributed residuals and homogeneous variances. The Kruskal-Wallis, BWS and Anderson-Darling omnibus test and all-pairs tests (Nemenyi test, Dunn test, Conover test, Dwass-Steele-Critchlow-Fligner test) as well as many-to-one (Nemenyi test, Dunn test, U-test) are given for the analysis of variance by ranks. Non-parametric trend tests (Jonckheere test, Cuzick test, Johnson-Mehrotra test, Spearman test) are included. In addition, a Friedman-test for one-way ANOVA with repeated measures on ranks (CRBD) and Skillings-Mack test for unbalanced CRBD is provided with consequent all-pairs tests (Nemenyi test, Siegel test, Miller test, Conover test, Exact test) and many-to-one tests (Nemenyi test, Demsar test, Exact test). A trend can be tested with Pages's test. Durbin's test for a two-way balanced incomplete block design (BIBD) is given in this package as well as Gore's test for CRBD with multiple observations per cell is given. Outlier tests, Mandel's k- and

h statistic as well as functions for Type I error and Power analysis as well as generic summary, print and plot methods are provided.

**Depends** R (>= 3.0.0)

**Imports** mvtnorm (>= 1.0), multcompView, gmp, Rmpfr, SuppDists, kSamples (>= 1.2.7), BWStest (>= 0.2.1), MASS

**Suggests** xtable, graphics, knitr, rmarkdown, car, e1071, multcomp, pwr

**SystemRequirements** gmp (>= 4.2.3), mpfr (>= 3.0.0) | file README

**VignetteBuilder** knitr, rmarkdown

**Classification/MSC-2010** 62J15, 62J10, 62G10, 62F03, 62G30

**NeedsCompilation** yes

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**License** GPL (>= 3)

**Repository** CRAN

**Date/Publication** 2018-01-17 09:42:40 UTC

## R topics documented:

adAllPairsTest . . . . .	4
adKSampleTest . . . . .	6
adManyOneTest . . . . .	7
algae . . . . .	9
barPlot . . . . .	10
bwsAllPairsTest . . . . .	11
bwsKSampleTest . . . . .	13
bwsManyOneTest . . . . .	15
bwsTrendTest . . . . .	17
chaAllPairsNashimotoTest . . . . .	19
chackoTest . . . . .	21
Cochran . . . . .	23
cochranTest . . . . .	24
cuzickTest . . . . .	26
Dgrubbs . . . . .	28
doubleGrubbsTest . . . . .	29
dscfAllPairsTest . . . . .	30
dunnettT3Test . . . . .	32
dunnettTest . . . . .	33
durbinAllPairsTest . . . . .	35
durbinTest . . . . .	37
frdAllPairsConoverTest . . . . .	38
frdAllPairsExactTest . . . . .	40
frdAllPairsMillerTest . . . . .	42

frdAllPairsNemenyiTest	44
frdAllPairsSiegelTest	46
frdManyOneDemsarTest	48
frdManyOneExactTest	50
frdManyOneNemenyiTest	52
friedmanTest	54
gamesHowellTest	56
gesdTest	57
goreTest	58
Grubbs	60
grubbsTest	61
johnsonTest	62
jonckheereTest	64
kruskalTest	66
kwAllPairsConoverTest	68
kwAllPairsDunnTest	70
kwAllPairsNemenyiTest	72
kwManyOneConoverTest	74
kwManyOneDunnTest	76
kwManyOneNdwTest	78
leTest	80
lsdTest	82
mackWolfeTest	84
Mandel-h	86
Mandel-k	87
mandelhTest	88
mandelkTest	90
manyOneUTest	91
MTest	93
normalScoresAllPairsTest	94
normalScoresManyOneTest	96
normalScoresTest	97
NPMTest	99
osrtTest	101
pageTest	102
Pentosan	103
plot.mandel	104
plot.PMCMR	104
powerMCTests	105
powerOneWayTests	107
print.gesdTest	109
print.mandel	109
print.PMCMR	110
print.powerOneWayPMCMR	110
print.powerPMCMR	111
qPCR	111
quadeAllPairsTest	112
reviewers	114

scheffeTest . . . . .	115
shirleyWilliamsTest . . . . .	116
skillingsMackTest . . . . .	119
spearmanTest . . . . .	120
summary.gesdTest . . . . .	122
summary.mandel . . . . .	123
summary.PMCMR . . . . .	123
summaryGroup . . . . .	124
tamhaneDunnettTest . . . . .	125
tamhaneT2Test . . . . .	126
trout . . . . .	128
tukeyTest . . . . .	129
uryWigginsHochbergTest . . . . .	130
vanWaerdenAllPairsTest . . . . .	132
vanWaerdenManyOneTest . . . . .	134
vanWaerdenTest . . . . .	136

<b>Index</b>	<b>138</b>
--------------	------------

---

adAllPairsTest	<i>Anderson-Darling All-Pairs Comparison Test</i>
----------------	---

---

## Description

Performs Anderson-Darling all-pairs comparison test.

## Usage

```
adAllPairsTest(x, ...)

## Default S3 method:
adAllPairsTest(x, g, p.adjust.method = p.adjust.methods,
  ...)

## S3 method for class 'formula'
adAllPairsTest(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.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).

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

### Details

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Anderson-Darling's all-pairs comparison test can be used. A total of  $m = k(k-1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : F_i(x) = F_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : F_i(x) \neq F_j(x)$ ,  $i \neq j$ .

This function is a wrapper function that sequentially calls `adKSampleTest` for each pair. The calculated p-values for  $\Pr(>|T2N|)$  can be adjusted to account for Type I error multiplicity using any method as implemented in `p.adjust`.

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

Scholz, F.W., Stephens, M.A. (1987) K-Sample Anderson-Darling Tests. *Journal of the American Statistical Association*, 82, 9118–924.

### See Also

[adKSampleTest](#), [adManyOneTest](#), [ad.pval](#).

**Examples**

```
adKSampleTest(count ~ spray, InsectSprays)

out <- adAllPairsTest(count ~ spray, InsectSprays, p.adjust="holm")
summary(out)
summaryGroup(out)
```

---

adKSampleTest	<i>Anderson-Darling k-Sample Test</i>
---------------	---------------------------------------

---

**Description**

Performs Anderson-Darling k-sample test.

**Usage**

```
adKSampleTest(x, ...)

## Default S3 method:
adKSampleTest(x, g, ...)

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

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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").

**Details**

The null hypothesis,  $H_0 : F_1 = F_2 = \dots = F_k$  is tested against the alternative,  $H_A : F_i \neq F_j$  ( $i \neq j$ ), with at least one unequality being strict.

This function only evaluates version 1 of the k-sample Anderson-Darling test (i.e. Eq. 6) of Scholz and Stephens (1987). The p-values are estimated with the extended empirical function as implemented in [ad.pval](#) of the package **kSamples**.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

Scholz, F.W., Stephens, M.A. (1987) K-Sample Anderson-Darling Tests. *Journal of the American Statistical Association*, 82, 9118–924.

**See Also**

[adAllPairsTest](#), [adManyOneTest](#), [ad.pval](#).

**Examples**

```
## Hollander & Wolfe (1973), 116.
## Mucociliary efficiency from the rate of removal of dust in normal
## subjects, subjects with obstructive airway disease, and subjects
## with asbestosis.
x <- c(2.9, 3.0, 2.5, 2.6, 3.2) # normal subjects
y <- c(3.8, 2.7, 4.0, 2.4)     # with obstructive airway disease
z <- c(2.8, 3.4, 3.7, 2.2, 2.0) # with asbestosis

datf <- data.frame(
  g = g <- c(rep("ns", length(x)), rep("oad",
    length(y)), rep("a", length(z))),
  x = x <- c(x, y, z))

adKSampleTest(x ~ g, datf)
```

---

adManyOneTest

*Anderson-Darling Many-To-One Comparison Test*


---

**Description**

Performs Anderson-Darling many-to-one comparison test.

**Usage**

```
adManyOneTest(x, ...)

## Default S3 method:
adManyOneTest(x, g, p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
adManyOneTest(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.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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> .

**Details**

For many-to-one comparisons (pairwise comparisons with one control) in an one-factorial layout with non-normally distributed residuals Anderson-Darling's non-parametric test can be performed. Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . Then  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : F_0 = F_i$  is tested in the two-tailed case against  $A_i : F_0 \neq F_i$ , ( $1 \leq i \leq m$ ).

This function is a wrapper function that sequentially calls `adKSampleTest` for each pair. The calculated p-values for  $\Pr(>|T2N|)$  can be adjusted to account for Type I error inflation using any method as implemented in [p.adjust](#).

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.



**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Scholz, F.W., Stephens, M.A. (1987) K-Sample Anderson-Darling Tests. *Journal of the American Statistical Association*, 82, 9118–924.

## See Also

[adKSampleTest](#), [adAllPairsTest](#), [ad.pval](#).

## Examples

```
## Data set PlantGrowth
## Global test
adKSampleTest(weight ~ group, data = PlantGrowth)

##
ans <- adManyOneTest(weight ~ group,
                     data = PlantGrowth,
                     p.adjust.method = "holm")
summary(ans)
```

---

algae

*Algae Growth Inhibition Data Set*

---

## Description

A dose-response experiment was conducted using Atrazine at 9 different dose-levels including the zero-dose control and the biomass of algae (*Selenastrum capricornutum*) as the response variable. Three replicates were measured at day 0, 1 and 2. The fluorescence method (Mayer et al. 1997) was applied to measure biomass.

## Format

A data frame with 22 observations on the following 10 variables.

**concentration** a numeric vector of dose value in mg / L

**Day.0** a numeric vector, total biomass

**Day.0.1** a numeric vector, total biomass

**Day.0.2** a numeric vector, total biomass

**Day.1** a numeric vector, total biomass

**Day.1.1** a numeric vector, total biomass

**Day.1.2** a numeric vector, total biomass

**Day.2** a numeric vector, total biomass

**Day.2.1** a numeric vector, total biomass

**Day.2.2** a numeric vector, total biomass

### Source

ENV/JM/MONO(2006)18/ANN, page 24.

### References

OECD (ed. 2006), *Current approaches in the statistical analysis of ecotoxicity data: A guidance to application - Annexes*. OECD Series on testing and assessment, No. 54, (ENV/JM/MONO(2006)18/ANN).

### See Also

demo(algae)

---

barPlot

*Plotting PMCMR Objects*

---

### Description

Plots a bar-plot for objects of class "PMCMR".

### Usage

```
barPlot(x, alpha = 0.05, ...)
```

### Arguments

x	an object of class "PMCMR".
alpha	the selected alpha-level. Defaults to 0.05.
...	further arguments for method barplot.

### Value

A barplot where the height of the bars corresponds to the arithmetic mean. The extend of the whiskers are  $\pm z_{(1-\alpha/2)} \times s_{E,i}$ , where the latter denotes the standard error of the *i*th group. Symbolic letters are depicted on top of the bars, whereas different letters indicate significant differences between groups for the selected level of alpha.

### Note

The barplot is strictly spoken only valid for normal data, as the depicted significance intervall implies symetry.

**Examples**

```
## data set chickwts
ans <- tukeyTest(weight ~ feed, data = chickwts)
barPlot(ans)
```

---

bwsAllPairsTest	<i>BWS All-Pairs Comparison Test</i>
-----------------	--------------------------------------

---

**Description**

Performs Baumgartner-Weiß-Schindler all-pairs comparison test.

**Usage**

```
bwsAllPairsTest(x, ...)

## Default S3 method:
bwsAllPairsTest(x, g, method = c("BWS", "Murakami"),
  p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
bwsAllPairsTest(formula, data, subset, na.action,
  method = c("BWS", "Murakami"), p.adjust.method = p.adjust.methods, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
method	a character string specifying the test statistic to use. Defaults to BWS.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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> .

## Details

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Baumgartner-Weiß-Schindler all-pairs comparison test can be used. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : F_i(x) = F_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : F_i(x) \neq F_j(x)$ ,  $i \neq j$ .

This function is a wrapper function that sequentially calls `{link[BWStest]}{bws_test}` for each pair. The default test method ("BWS") is the original Baumgartner-Weiß-Schindler test statistic B. For method == "Murakami" it is the modified BWS statistic denoted B\*. The calculated p-values for  $\Pr(>|B|)$  or  $\Pr(>|B^*|)$  can be adjusted to account for Type I error inflation using any method as implemented in `p.adjust`.

## Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Baumgartner, W., Weiss, P., Schindler, H. (1998), A nonparametric test for the general two-sample problem, *Biometrics* 54, 1129–1135.

Murakami, H. (2006) K-sample rank test based on modified Baumgartner statistic and its power comparison, *J. Jpn. Comp. Statist.* 19, 1–13.

## See Also

`bws_test`.

## Examples

```
out <- bwsAllPairsTest(count ~ spray, InsectSprays, p.adjust="holm")
summary(out)
summaryGroup(out)
```

---

bwsKSTest	<i>Murakami's k-Sample BWS Test</i>
-----------	-------------------------------------

---

## Description

Performs Murakami's k-Sample BWS Test.

## Usage

```
bwsKSTest(x, ...)

## Default S3 method:
bwsKSTest(x, g, nperm = 1000, ...)

## S3 method for class 'formula'
bwsKSTest(formula, data, subset, na.action,
           nperm = 1000, ...)
```

## Arguments

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>nperm</code>	number of permutations for the asymptotic permutation test. Defaults to 1000.
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<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> .

## Details

The null hypothesis,  $H_0 : F_1 = F_2 = \dots = F_k$  is tested against the alternative,  $H_A : F_i \neq F_j$  ( $i \neq j$ ), with at least one inequality being strict.

The p-values are estimated through an asymptotic boot-strap method.

**Value**

A list with class "hstest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**Note**

One may increase the number of permutations to e.g. `nperm = 10000` in order to get more precise p-values. However, this will be on the expense of computational time.

**References**

Murakami, H. (2006) K-sample rank test based on modified Baumgartner statistic and its power comparison, *J. Jpn. Comp. Statist.* 19, 1–13.

**See Also**

[sample](#), [bwsAllPairsTest](#), [link{bwsManyOneTest}](#).

**Examples**

```
#' ## Hollander & Wolfe (1973), 116.
## Mucociliary efficiency from the rate of removal of dust in normal
## subjects, subjects with obstructive airway disease, and subjects
## with asbestosis.
x <- c(2.9, 3.0, 2.5, 2.6, 3.2) # normal subjects
y <- c(3.8, 2.7, 4.0, 2.4)     # with obstructive airway disease
z <- c(2.8, 3.4, 3.7, 2.2, 2.0) # with asbestosis

datf <- data.frame(
  g = g <- c(rep("ns", length(x)), rep("oad",
    length(y)), rep("a", length(z))),
  x = x <- c(x, y, z))

## k-sample BWS Test
bwsKSampleTest(x ~ g, datf)
```

---

bwsManyOneTest	<i>BWS Many-To-One Comparison Test</i>
----------------	--

---

## Description

Performs Baumgartner-Weiß-Schindler many-to-one comparison test.

## Usage

```
bwsManyOneTest(x, ...)

## Default S3 method:
bwsManyOneTest(x, g, alternative = c("two.sided", "greater",
  "less"), method = c("BWS", "Murakami", "Neuhauser"),
  p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
bwsManyOneTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), method = c("BWS",
  "Murakami", "Neuhauser"), p.adjust.method = p.adjust.methods, ...)
```

## Arguments

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>alternative</code>	the alternative hypothesis. Defaults to <code>two.sided</code> .
<code>method</code>	a character string specifying the test statistic to use. Defaults to <code>BWS</code> .
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

## Details

For many-to-one comparisons (pairwise comparisons with one control) in an one-factorial layout with non-normally distributed residuals Baumgartner-Weiß-Schindler's non-parametric test can be performed. Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . Then  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : F_0 = F_i$  is tested in the two-tailed case against  $A_i : F_0 \neq F_i$ , ( $1 \leq i \leq m$ ).

This function is a wrapper function that sequentially calls `{link[BWStest]}{bws_stat}` and `{link[BWStest]}{bws_cdf}` for each pair. For the default test method ("BWS") the original Baumgartner-Weiß-Schindler test statistic  $B$  and its corresponding  $\Pr(>|B|)$  is calculated. For method == "BWS" only a two-sided test is possible.

For method == "Murakami" the modified BWS statistic denoted  $B^*$  and its corresponding  $\Pr(>|B^*|)$  is computed by sequentially calling `{link[BWStest]}{murakami_stat}` and `{link[BWStest]}{murakami_cdf}`. For method == "Murakami" only a two-sided test is possible.

If alternative == "greater" then the alternative, if one population is stochastically larger than the other is tested:  $H_i : F_0 = F_i$  against  $A_i : F_0 \geq F_i$ , ( $1 \leq i \leq m$ ). The modified test-statistic  $B^*$  according to Neuhäuser (2001) and its corresponding  $\Pr(>B^*)$  or  $\Pr(<B^*)$  is computed by sequentially calling `{link[BWStest]}{murakami_stat}` and `{link[BWStest]}{murakami_cdf}` with `flavor = 2`.

The p-values can be adjusted to account for Type I error inflation using any method as implemented in [p.adjust](#).

## Value

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for p-value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

## References

- Baumgartner, W., Weiss, P., Schindler, H. (1998), A nonparametric test for the general two-sample problem, *Biometrics* 54, 1129–1135.
- Murakami, H. (2006) K-sample rank test based on modified Baumgartner statistic and its power comparison, *J. Jpn. Comp. Statist.* 19, 1–13.
- Neuhäuser, M. (2001) One-side two-sample and trend tests based on a modified Baumgartner-Weiss-Schindler statistic. *Journal of Nonparametric Statistics*, 13, 729–739.

## See Also

[murakami\\_stat](#), [murakami\\_cdf](#), [bws\\_stat](#), [bws\\_cdf](#).



**Examples**

```

out <- bwsManyOneTest(weight ~ group, PlantGrowth, p.adjust="holm")
summary(out)

## A two-sample test
set.seed(1245)
x <- c(rnorm(20), rnorm(20,0.3))
g <- gl(2, 20)
summary(bwsManyOneTest(x ~ g, alternative = "less", p.adjust="none"))
summary(bwsManyOneTest(x ~ g, alternative = "greater", p.adjust="none"))

## Not run:
## Check with the implementation in package BWStest
BWStest::bws_test(x=x[g==1], y=x[g==2], alternative = "less")
BWStest::bws_test(x=x[g==1], y=x[g==2], alternative = "greater")

## End(Not run)

```

---

bwsTrendTest

*Testing against Ordered Alternatives (Murakami's BWS Trend Test)*


---

**Description**

Performs Murakami's modified Baumgartner-Weiß-Schindler test for testing against ordered alternatives.

**Usage**

```

bwsTrendTest(x, ...)

## Default S3 method:
bwsTrendTest(x, g, nperm = 1000, ...)

## S3 method for class 'formula'
bwsTrendTest(formula, data, subset, na.action, nperm = 1000,
  ...)

```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
nperm	number of permutations for the asymptotic permutation test. Defaults to 1000.
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 <a href="#">model.frame</a> ) 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> .

### Details

The null hypothesis,  $H_0 : F_1(u) = F_2(u) = \dots = F_k(u) \quad u \in R$  is tested against a simple order hypothesis,  $H_A : F_1(u) \leq F_2(u) \leq \dots \leq F_k(u), F_1(u) < F_k(u), \quad u \in R$ .

The p-values are estimated through an asymptotic boot-strap method using the function [sample](#).

### Value

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

### Note

One may increase the number of permutations to e.g. `nperm = 10000` in order to get more precise p-values. However, this will be on the expense of computational time.

### References

Murakami, H. (2006) K-sample rank test based on modified Baumgartner statistic and its power comparison, *J. Jpn. Comp. Statist.* 19, 1–13.

Neuhäuser, M. (2001) One-side two-sample and trend tests based on a modified Baumgartner-Weiss-Schindler statistic. *Journal of Nonparametric Statistics*, 13, 729–739.

### See Also

[sample](#), [bwsAllPairsTest](#), [link{bwsManyOneTest}](#).

[kruskalTest](#) and [shirleyWilliamsTest](#) of the package **PMCMRplus**, [kruskal.test](#) of the library **stats**.

**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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)
```

---

chaAllPairsNashimotoTest

*All-Pairs Comparisons for Simply Ordered Mean Ranksums*


---

**Description**

Performs Nashimoto and Wright's all-pairs comparison procedure for simply ordered mean ranksums (NPY' test and NPT' test). According to the authors, bove procedures shall only be applied after Chacko's test (see [chackoTest](#) indicates global significance).

The modified procedure uses the property of a simple order,  $\theta'_m - \theta_m \leq \theta_j - \theta_i \leq \theta'_l - \theta_l$  ( $l \leq i \leq m$  and  $m' \leq j \leq l'$ ). The null hypothesis  $H_{ij} : \theta_i = \theta_j$  is tested against the alternative  $A_{ij} : \theta_i < \theta_j$  for any  $1 \leq i < j \leq k$ .

In the NPY' test the p-values are estimated from the studentized range distribution. In the NPT' test the p-values are estimated from the standard normal distribution.

**Usage**

```

chaAllPairsNashimotoTest(x, ...)

## Default S3 method:
chaAllPairsNashimotoTest(x, g,
  p.adjust.method = c("single-step", p.adjust.methods), ...)

## S3 method for class 'formula'
chaAllPairsNashimotoTest(formula, data, subset, na.action,
  p.adjust.method = c("single-step", p.adjust.methods), ...)

```

**Arguments**

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

**Details**

The type of test can be controlled via the argument `p.adjust.method`:

**single.step** the NPY' test is performed.

**none** the plain NPT' test is performed.

However, any method as available by [p.adjust.methods](#) can be selected for the adjustment of p-values estimated from the standard normal distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Nashimoto, K., Wright, F.T., (2005), Multiple comparison procedures for detecting differences in simply ordered means. *Comput. Statist. Data Anal.* 48, 291–306.

## See Also

[Tukey](#), [Normal](#), [chackoTest](#)

## 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 <- gl(3,5)
levels(g) <- c("A", "B", "C")
chackoTest(x, g)
chaAllPairsNashimotoTest(x, g, p.adjust.method = "single-step")
```

---

chackoTest

*Testing against Ordered Alternatives (Chacko's Test)*

---

## Description

Performs Chacko's test for testing against ordered alternatives.

## Usage

```
chackoTest(x, ...)

## Default S3 method:
chackoTest(x, g, ...)

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

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <code>model.frame</code> ) 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> .

**Details**

The null hypothesis,  $H_0 : \theta_1 = \theta_2 = \dots = \theta_k$  is tested against a simple order hypothesis,  $H_A : \theta_1 \leq \theta_2 \leq \dots \leq \theta_k, \theta_1 < \theta_k$ .

The p-values are estimated from the chi-square distribution.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**Source**

The source code for the application of the pool adjacent violators theorem to calculate the isotonic means was taken from the file "pava.f", which is included in the package **Iso**:

Rolf Turner (2015). Iso: Functions to Perform Isotonic Regression. R package version 0.0-17. <https://CRAN.R-project.org/package=Iso>.

The file "pava.f" is a Ratfor modification of Algorithm AS 206.1:

Bril, G., Dykstra, R., Pillers, C., Robertson, T. (1984) Statistical Algorithms: Algorithm AS 206: Isotonic Regression in Two Independent Variables, *Appl. Statist.*, 34, 352–357.

The Algorithm AS 206 is available from StatLib <http://lib.stat.cmu.edu/apstat>. The Royal Statistical Society holds the copyright to these routines, but has given its permission for their distribution provided that no fee is charged.

## References

Chacko, V. J. (1963), Testing homogeneity against ordered alternatives. *Ann. Math. Statist.*, 34, 945–956.

## See Also

[kruskalTest](#) and [shirleyWilliamsTest](#) of the package [PMCMRplus](#), [kruskal.test](#) of the library [stats](#).

## 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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)
```

---

Cochran

*Cochran's distribution*

---

## Description

Distribution function and quantile function for Cochran's distribution.

**Usage**

```
qcochran(p, k, n, lower.tail = TRUE, log.p = FALSE)
```

```
pcochran(q, k, n, lower.tail = TRUE, log.p = FALSE)
```

**Arguments**

p	vector of probabilities.
k	number of groups.
n	(average) sample size of the k groups.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .
log.p	logical; if TRUE, probabilities p are given as $\log(p)$ .
q	vector of quantiles.

**Value**

pcochran gives the distribution function and qcochran gives the quantile function.

**References**

Cochran, W.G. (1941) The distribution of the largest of a set of estimated variances as a fraction of their total. *Ann. Eugen.* 11, 47–52.

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.* <http://dx.doi.org/10.1007/s10182-011-0185-y>.

**See Also**

[FDist](#)

**Examples**

```
qcochran(0.05, 7, 3)
```

---

cochranTest

*Cochran Test*

---

**Description**

Performs Cochran's test for testing an outlying (or inlying) variance.



**Usage**

```

cochranTest(x, ...)

## Default S3 method:
cochranTest(x, g, alternative = c("greater", "less"), ...)

## S3 method for class 'formula'
cochranTest(formula, data, subset, na.action,
             alternative = c("greater", "less"), ...)

```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "greater"
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 <a href="#">model.frame</a> ) 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").

**Details**

For normally distributed data the null hypothesis,  $H_0 : \sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2$  is tested against the alternative (greater)  $H_A : \sigma_p > \sigma_i$  ( $i \leq k, i \neq p$ ) with at least one inequality being strict.

The p-value is computed with the function [pcochran](#).

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

Cochran, W.G. (1941) The distribution of the largest of a set of estimated variances as a fraction of their total. *Ann. Eugen.* 11, 47–52.

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.* <http://dx.doi.org/10.1007/s10182-011-0185-y>.

**See Also**

[bartlett.test](#), [fligner.test](#).

**Examples**

```
data(Pentosan)
cochranTest(value ~ lab, data = Pentosan, subset = (material == "A"))
```

---

cuzickTest

*Testing against Ordered Alternatives (Cuzick's Test)*


---

**Description**

Performs Cuzick's test for testing against ordered alternatives.

**Usage**

```
cuzickTest(x, ...)

## Default S3 method:
cuzickTest(x, g, alternative = c("two.sided", "greater",
  "less"), scores = NULL, continuity = FALSE, ...)

## S3 method for class 'formula'
cuzickTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), scores = NULL,
  continuity = FALSE, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "two.sided".
scores	numeric vector of scores. Defaults to NULL.
continuity	logical indicator whether a continuity correction shall be performed. Defaults to FALSE.

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

### Details

The null hypothesis,  $H_0 : \theta_1 = \theta_2 = \dots = \theta_k$  is tested against a simple order hypothesis,  $H_A : \theta_1 \leq \theta_2 \leq \dots \leq \theta_k, \theta_1 < \theta_k$ .

The p-values are estimated from the standard normal distribution.

### Value

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

### References

Cuzick, J. (1995). A wilcoxon-type test for trend. *Statistics in Medicine*, 4, 87–90.

### See Also

`kruskalTest` and `shirleyWilliamsTest` of the package `PMCMRplus`, `kruskal.test` of the library `stats`.

### 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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
```

```

chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)

```

---

Dgrubbs

*Grubbs D\* distribution*


---

### Description

Distribution function for Grubbs D\* distribution.

### Usage

```
pdgrubbs(q, n, m = 10000, lower.tail = TRUE, log.p = FALSE)
```

### Arguments

q	vector of quantiles.
n	total sample size.
m	number of Monte-Carlo replicates. Defaults to 10,000.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .
log.p	logical; if TRUE, probabilities p are given as $\log(p)$ .

### Value

pgrubbs gives the distribution function

### References

Grubbs, F.E. (1950) Sample criteria for testing outlying observations. *Ann. Math. Stat.* 21, 27–58.  
 Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.* <http://dx.doi.org/10.1007/s10182-011-0185-y>.

**See Also**[Grubbs](#)**Examples**

```
pdgrubbs(0.62, 7, 1E4)
```

---

doubleGrubbsTest	<i>Grubbs Double Outlier Test</i>
------------------	-----------------------------------

---

**Description**

Performs Grubbs double outlier test.

**Usage**

```
doubleGrubbsTest(x, alternative = c("two.sided", "greater", "less"),
  m = 10000)
```

**Arguments**

<code>x</code>	a numeric vector of data.
<code>alternative</code>	the alternative hypothesis. Defaults to "two.sided".
<code>m</code>	number of Monte-Carlo replicates.

**Details**

Let  $X$  denote an identically and independently distributed continuous variate with realizations  $x_i$  ( $1 \leq i \leq n$ ). Further, let the increasingly ordered realizations denote  $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$ . Then the following model for testing two maximum outliers can be proposed:

$$x_{(i)} = \begin{cases} \mu + \epsilon_{(i)}, & i = 1, \dots, n-2 \\ \mu + \Delta + \epsilon_{(j)}, & j = n-1, n \end{cases}$$

with  $\epsilon \approx N(0, \sigma)$ . The null hypothesis,  $H_0 : \Delta = 0$  is tested against the alternative,  $H_A : \Delta > 0$ .

For testing two minimum outliers, the model can be proposed as

$$x_{(i)} = \begin{cases} \mu + \Delta + \epsilon_{(j)}, & j = 1, 2 \\ \mu + \epsilon_{(i)}, & i = 3, \dots, n \end{cases}$$

The null hypothesis is tested against the alternative,  $H_A : \Delta < 0$ .

The p-value is computed with the function [pdgrubbs](#).

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

Grubbs, F.E. (1950) Sample criteria for testing outlying observations. *Ann. Math. Stat.* 21, 27–58.

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.* <http://dx.doi.org/10.1007/s10182-011-0185-y>.

**Examples**

```
data(Pentosan)
dat <- subset(Pentosan, subset = (material == "A"))
labMeans <- tapply(dat$value, dat$lab, mean)
doubleGrubbsTest(x = labMeans, alternative = "less")
```

---

dscfAllPairsTest      *Multiple Comparisons of Mean Rank Sums*

---

**Description**

Performs the all-pairs comparison test for different factor levels according to Dwass, Steel, Critchlow and Fligner.

**Usage**

```
dscfAllPairsTest(x, ...)
```

## Default S3 method:

```
dscfAllPairsTest(x, g, ...)
```

## S3 method for class 'formula'

```
dscfAllPairsTest(formula, data, subset, na.action, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <code>model.frame</code> ) 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals the DSCF all-pairs comparison test can be used. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : F_i(x) = F_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : F_i(x) \neq F_j(x)$ ,  $i \neq j$ . As opposed to the all-pairs comparison procedures that depend on Kruskal ranks, the DSCF test is basically an extension of the U-test as re-ranking is conducted for each pairwise test.

The p-values are estimated from the studentized range distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

C. E. Douglas, A. M. Fligner A. (1991) On distribution-free multiple comparisons in the one-way analysis of variance, *Communications in Statistics - Theory and Methods*, 20, 127–139.

M. Dwass (1960) Some k-sample rank-order tests. In *Contributions to Probability and Statistics*, Edited by: I. Olkin, Stanford: Stanford University Press.

R. G. D. Steel (1960) A rank sum test for comparing all pairs of treatments, *Technometrics*, 2, 197–207

**See Also**

[Tukey, pairwise.wilcox.test](#)

---

dunnettT3Test                      *Dunnett's T3 Test*

---

**Description**

Performs Dunnett's all-pairs comparison test for normally distributed data with unequal variances.

**Usage**

```
dunnettT3Test(x, ...)

## Default S3 method:
dunnettT3Test(x, g, ...)

## S3 method for class 'formula'
dunnettT3Test(formula, data, subset, na.action, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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").

**Details**

For all-pairs comparisons in an one-factorial layout with normally distributed residuals but unequal groups variances the T3 test of Dunnett can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

The p-values are computed from the studentized maximum modulus distribution that is the equivalent of the multivariate t distribution with  $\rho = 0$ . The function [pmvt](#) is used to calculate the p-values.



**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

C. W. Dunnett (1980) Pair wise multiple comparisons in the unequal variance case. *Journal of the American Statistical Association*, 75, 796–800.

**See Also**

[pmvt](#)

**Examples**

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1:5, each=5)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 != varN
anova(fit)
summary(dunnettT3Test(x, g))
```

---

dunnettTest

*Dunnett's Many-to-One Comparisons Test*

---

**Description**

Performs Dunnett's multiple comparisons test with one control.

**Usage**

```
dunnettTest(x, ...)

## Default S3 method:
dunnettTest(x, g, alternative = c("two.sided", "greater",
  "less"), ...)

## S3 method for class 'formula'
dunnettTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), ...)
```

**Arguments**

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>alternative</code>	the alternative hypothesis. Defaults to <code>two.sided</code> .
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

**Details**

For many-to-one comparisons in an one-factorial layout with normally distributed residuals Dunnett's test can be used. A total of  $m = k - 1$  hypotheses can be tested. The null hypothesis  $H_i : \mu_0(x) = \mu_i(x)$  is tested in the two-tailed test against the alternative  $A_i : \mu_0(x) \neq \mu_i(x)$ ,  $1 \leq i \leq k - 1$ .

The p-values for the test are calculated from the multivariate t distribution as implemented in the function [pmvt](#).

**Value**

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

OECD (ed. 2006), *Current approaches in the statistical analysis of ecotoxicity data: A guidance to application - Annexes*. OECD Series on testing and assessment, No. 54.

## See Also

[pmvt](#)

## Examples

```
set.seed(245)
mn <- c(1, 2, 2^2, 2^3, 2^4)
x <- rep(mn, each=5) + rnorm(25)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g - 1)
shapiro.test(residuals(fit))
bartlett.test(x ~ g - 1)
anova(fit)
summary(dunnettTest(x, g, alternative = "greater"))
```

---

`durbinAllPairsTest`      *All-Pairs Comparisons Test for Balanced Incomplete Block Designs*

---

## Description

Performs Conover-Iman all-pairs comparison test for a balanced incomplete block design (BIBD).

## Usage

```
durbinAllPairsTest(y, ...)

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

**Arguments**

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> )
<code>...</code>	further arguments to be passed to or from methods.

**Details**

For all-pairs comparisons in a balanced incomplete block design the proposed test of Conover and Imam can be applied. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \theta_i = \theta_j$  is tested in the two-tailed test against the alternative  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

The p-values are computed from the t distribution. If no p-value adjustment is performed (`p.adjust.method = "none"`), than a simple protected test is recommended, i.e. the all-pairs comparisons should only be applied after a significant [durbinTest](#). However, any method as implemented in [p.adjust.methods](#) can be selected by the user.

**Value**

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for p-value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

**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**

[durbinTest](#)

**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]))
durbinAllPairsTest(y)
```

---

durbinTest

*Durbin Test*


---

**Description**

Performs Durbin's tests whether  $k$  groups (or treatments) in a two-way balanced incomplete block design (BIBD) have identical effects.

**Usage**

```
durbinTest(y, ...)

## Default S3 method:
durbinTest(y, groups, blocks, ...)
```

**Arguments**

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>...</code>	further arguments to be passed to or from methods.

**Details**

For testing a two factorial layout of a balanced incomplete block design whether the  $k$  groups have identical effects, the Durbin test can be performed. The null hypothesis,  $H_0 : \theta_i = \theta_j$  ( $1 \leq i < j \leq k$ ), is tested against the alternative that at least one  $\theta_i \neq \theta_j$ .

The p-values are computed from the chi-square distribution.

**Value**

A list with class "hstest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**Note**

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

**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.

**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, 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]))
durbinTest(y)
```

---

frdAllPairsConoverTest

*Conover's All-Pairs Comparisons Test for Unreplicated Blocked Data*

---

**Description**

Performs Conover's all-pairs comparisons tests of Friedman-type ranked data.

**Usage**

```
frdAllPairsConoverTest(y, ...)

## Default S3 method:
frdAllPairsConoverTest(y, groups, blocks,
  p.adjust.method = c("single-step", p.adjust.methods), ...)
```

**Arguments**

**y** a numeric vector of data values, or a list of numeric data vectors.

**groups** a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.

**blocks** a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning 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 all-pairs comparisons in a two factorial unreplicated complete block design with non-normally distributed residuals, Conover's test can be performed on Friedman-type ranked data.

A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis,  $H_{ij} : \theta_i = \theta_j$ , is tested in the two-tailed case against the alternative,  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

If `p.adjust.method == "single-step"` the p-values are computed from the studentized range distribution. Otherwise, the p-values are computed from the t-distribution using any of the p-adjustment methods as included in [p.adjust](#).

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**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**

[friedmanTest](#), [friedman.test](#), [frdAllPairsExactTest](#), [frdAllPairsMillerTest](#), [frdAllPairsNemenyiTest](#), [frdAllPairsSiegelTest](#)

**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, LETTERS[1:6]))
print(y)
friedmanTest(y)

## Eisinga et al. 2017
frdAllPairsExactTest(y=y, p.adjust = "bonferroni")

## Conover's test
frdAllPairsConoverTest(y=y, p.adjust = "bonferroni")

## Nemenyi's test
frdAllPairsNemenyiTest(y=y)

## Miller et al.
frdAllPairsMillerTest(y=y)
```

---

frdAllPairsExactTest *Exact All-Pairs Comparisons Test for Unreplicated Blocked Data*

---

**Description**

Performs exact all-pairs comparisons tests of Friedman-type ranked data according to Eisinga et al. (2017).

**Usage**

```
frdAllPairsExactTest(y, ...)

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



**Arguments**

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>...</code>	further arguments to be passed to or from methods.

**Details**

For all-pairs comparisons in a two factorial unreplicated complete block design with non-normally distributed residuals, an exact test can be performed on Friedman-type ranked data.

A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis,  $H_{ij} : \theta_i = \theta_j$ , is tested in the two-tailed case against the alternative,  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

The exact  $p$ -values are computed using the code of "pexactfrsd.R" that was a supplement to the publication of Eisinga et al. (2017). Additionally, any of the  $p$ -adjustment methods as included in [p.adjust](#) can be selected, for  $p$ -value adjustment.

**Value**

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the  $p$ -values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for  $p$ -value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

**Source**

The function `frdAllPairsExactTest` uses the code of the file `pexactfrsd.R` that was a supplement to:

R. Eisinga, T. Heskes, B. Pelzer, M. Te Grotenhuis (2017), Exact  $p$ -values for Pairwise Comparison of Friedman Rank Sums, with Application to Comparing Classifiers, *BMC Bioinformatics*, 18:68.

**References**

R. Eisinga, T. Heskes, B. Pelzer, M. Te Grotenhuis (2017), Exact  $p$ -values for Pairwise Comparison of Friedman Rank Sums, with Application to Comparing Classifiers, *BMC Bioinformatics*, 18:68.

**See Also**

[friedmanTest](#), [friedman.test](#), [frdAllPairsConoverTest](#), [frdAllPairsMillerTest](#), [frdAllPairsNemenyiTest](#), [frdAllPairsSiegelTest](#)

**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, LETTERS[1:6]))
print(y)
friedmanTest(y)

## Eisinga et al. 2017
frdAllPairsExactTest(y=y, p.adjust = "bonferroni")

## Conover's test
frdAllPairsConoverTest(y=y, p.adjust = "bonferroni")

## Nemenyi's test
frdAllPairsNemenyiTest(y=y)

## Miller et al.
frdAllPairsMillerTest(y=y)
```

---

frdAllPairsMillerTest *Millers's All-Pairs Comparisons Test for Unreplicated Blocked Data*

---

**Description**

Performs Miller's all-pairs comparisons tests of Friedman-type ranked data.

**Usage**

```
frdAllPairsMillerTest(y, ...)

## Default S3 method:
frdAllPairsMillerTest(y, groups, blocks, ...)
```

**Arguments**

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>...</code>	further arguments to be passed to or from methods.

**Details**

For all-pairs comparisons in a two factorial unreplicated complete block design with non-normally distributed residuals, Miller's test can be performed on Friedman-type ranked data.

A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis,  $H_{ij} : \theta_i = \theta_j$ , is tested in the two-tailed case against the alternative,  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

The  $p$ -values are computed from the chi-square distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the  $p$ -values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for  $p$ -value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

J. Bortz J, G. A. Lienert, K. Boehnke (1990), *Verteilungsfreie Methoden in der Biostatistik*. Berlin: Springer.

R. G. Miller Jr. (1996), *Simultaneous statistical inference*. New York: McGraw-Hill.

E. L. Wike (2006), *Data Analysis. A Statistical Primer for Psychology Students*. New Brunswick: Aldine Transaction.

**See Also**

[friedmanTest](#), [friedman.test](#), [frdAllPairsExactTest](#), [frdAllPairsConoverTest](#), [frdAllPairsNemenyiTest](#), [frdAllPairsSiegelTest](#)

**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, LETTERS[1:6]))
print(y)
friedmanTest(y)

## Eisinga et al. 2017
frdAllPairsExactTest(y=y, p.adjust = "bonferroni")

## Conover's test
frdAllPairsConoverTest(y=y, p.adjust = "bonferroni")

## Nemenyi's test
frdAllPairsNemenyiTest(y=y)

## Miller et al.
frdAllPairsMillerTest(y=y)

```

---

frdAllPairsNemenyiTest

*Nemenyi's All-Pairs Comparisons Test for Unreplicated Blocked Data*


---

**Description**

Performs Nemenyi's all-pairs comparisons tests of Friedman-type ranked data.

**Usage**

```

frdAllPairsNemenyiTest(y, ...)

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

```

**Arguments**

**y** a numeric vector of data values, or a list of numeric data vectors.

**groups** a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.

blocks a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.

... further arguments to be passed to or from methods.

### Details

For all-pairs comparisons in a two factorial unreplicated complete block design with non-normally distributed residuals, Nemenyi's test can be performed on Friedman-type ranked data.

A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis,  $H_{ij} : \theta_i = \theta_j$ , is tested in the two-tailed case against the alternative,  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

The  $p$ -values are computed from the studentized range distribution.

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the  $p$ -values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for  $p$ -value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

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

R. G. Miller Jr. (1996), *Simultaneous statistical inference*. New York: McGraw-Hill.

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

L. Sachs (1997), *Angewandte Statistik*. Berlin: Springer. Pages: 668–675.

### See Also

[friedmanTest](#), [friedman.test](#), [frdAllPairsExactTest](#), [frdAllPairsConoverTest](#), [frdAllPairsMillerTest](#), [frdAllPairsSiegelTest](#)

### 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, LETTERS[1:6]))
print(y)
friedmanTest(y)

## Eisinga et al. 2017
frdAllPairsExactTest(y=y, p.adjust = "bonferroni")

## Conover's test
frdAllPairsConoverTest(y=y, p.adjust = "bonferroni")

## Nemenyi's test
frdAllPairsNemenyiTest(y=y)

## Miller et al.
frdAllPairsMillerTest(y=y)

```

---

frdAllPairsSiegelTest *Siegel and Castellan's All-Pairs Comparisons Test for Unreplicated Blocked Data*

---

## Description

Performs Siegel and Castellan's all-pairs comparisons tests of Friedman-type ranked data.

## Usage

```

frdAllPairsSiegelTest(y, ...)

## Default S3 method:
frdAllPairsSiegelTest(y, groups, blocks,
  p.adjust.method = p.adjust.methods, ...)

```

## Arguments

y	a numeric vector of data values, or a list of numeric data vectors.
groups	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
blocks	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
...	further arguments to be passed to or from methods.

## Details

For all-pairs comparisons in a two factorial unreplicated complete block design with non-normally distributed residuals, Siegel and Castellan's test can be performed on Friedman-type ranked data.

A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis,  $H_{ij} : \theta_i = \theta_j$ , is tested in the two-tailed case against the alternative,  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

The  $p$ -values are computed from the standard normal distribution. Any method as implemented in [p.adjust](#) can be used for  $p$ -value adjustment.

## Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the  $p$ -values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for  $p$ -value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

S. Siegel, N. J. Castellan Jr. (1988), *Nonparametric Statistics for the Behavioral Sciences*. 2nd ed. New York: McGraw-Hill.

## See Also

[friedmanTest](#), [friedman.test](#), [frdAllPairsExactTest](#), [frdAllPairsConoverTest](#), [frdAllPairsNemenyiTest](#), [frdAllPairsMillerTest](#)

## 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, LETTERS[1:6]))
print(y)
friedmanTest(y)
```

```
## Eisinga et al. 2017
frdAllPairsExactTest(y=y, p.adjust = "bonferroni")

## Conover's test
frdAllPairsConoverTest(y=y, p.adjust = "bonferroni")

## Nemenyi's test
frdAllPairsNemenyiTest(y=y)

## Miller et al.
frdAllPairsMillerTest(y=y)
```

---

frdManyOneDemsarTest *Demsar's Many-to-One Test for Unreplicated Blocked Data*

---

## Description

Performs Demsar's non-parametric many-to-one comparison test for Friedman-type ranked data.

## Usage

```
frdManyOneDemsarTest(y, ...)

## Default S3 method:
frdManyOneDemsarTest(y, groups, blocks,
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = p.adjust.methods, ...)
```

## Arguments

y	a numeric vector of data values, or a list of numeric data vectors.
groups	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
blocks	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
...	further arguments to be passed to or from methods.

## Details

For many-to-one comparisons (pairwise comparisons with one control) in a two factorial unreplicated complete block design with non-normally distributed residuals, Demsar's test can be performed on Friedman-type ranked data.



Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . A total of  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : \theta_0 = \theta_i$  is tested in the two-tailed case against  $A_i : \theta_0 \neq \theta_i$ , ( $1 \leq i \leq m$ ).

The  $p$ -values are computed from the standard normal distribution. Any of the  $p$ -adjustment methods as included in [p.adjust](#) can be used for the adjustment of  $p$ -values.

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the  $p$ -values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for  $p$ -value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

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

### See Also

[friedmanTest](#), [friedman.test](#), [frdManyOneExactTest](#), [frdManyOneNemenyiTest](#).

### 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.
## Assume A is the control.

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]))

## Global Friedman test
friedmanTest(y)

## Demsar's many-one test
```

```

frdManyOneDemsarTest(y=y, p.adjust = "bonferroni")

## Exact many-one test
frdManyOneExactTest(y=y, p.adjust = "bonferroni")

## Nemenyi's many-one test
frdManyOneNemenyiTest(y=y)

```

---

```
frdManyOneExactTest Exact Many-to-One Test for Unreplicated Blocked Data
```

---

### Description

Performs an exact non-parametric many-to-one comparison test for Friedman-type ranked data according to Eisinga et al. (2017).

### Usage

```

frdManyOneExactTest(y, ...)

## Default S3 method:
frdManyOneExactTest(y, groups, blocks,
  p.adjust.method = p.adjust.methods, ...)

```

### Arguments

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>...</code>	further arguments to be passed to or from methods.

### Details

For many-to-one comparisons (pairwise comparisons with one control) in a two factorial unreplicated complete block design with non-normally distributed residuals, an exact test can be performed on Friedman-type ranked data.

Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . A total of  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : \theta_0 = \theta_i$  is tested in the two-tailed case against  $A_i : \theta_0 \neq \theta_i$ , ( $1 \leq i \leq m$ ).

The exact  $p$ -values are computed using the code of "pexactfrsd.R" that was a supplement to the publication of Eisinga et al. (2017). Additionally, any of the  $p$ -adjustment methods as included in [p.adjust](#) can be selected, for  $p$ -value adjustment.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

R. Eisinga, T. Heskes, B. Pelzer, M. Te Grotenhuis (2017), Exact p-values for Pairwise Comparison of Friedman Rank Sums, with Application to Comparing Classifiers, *BMC Bioinformatics*, 18:68.

**See Also**

[friedmanTest](#), [friedman.test](#), [frdManyOneDemsarTest](#), [frdManyOneNemenyiTest](#).

**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.
## Assume A is the control.

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]))

## Global Friedman test
friedmanTest(y)

## Demsar's many-one test
frdManyOneDemsarTest(y=y, p.adjust = "bonferroni")

## Exact many-one test
frdManyOneExactTest(y=y, p.adjust = "bonferroni")

## Nemenyi's many-one test
frdManyOneNemenyiTest(y=y)
```

---

frdManyOneNemenyiTest *Nemenyi's Many-to-One Test for Unreplicated Blocked Data*

---

### Description

Performs Nemenyi's non-parametric many-to-one comparison test for Friedman-type ranked data.

### Usage

```
frdManyOneNemenyiTest(y, ...)

## Default S3 method:
frdManyOneNemenyiTest(y, groups, blocks,
  alternative = c("two.sided", "greater", "less"), ...)
```

### Arguments

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>alternative</code>	the alternative hypothesis. Defaults to <code>two.sided</code> .
<code>...</code>	further arguments to be passed to or from methods.

### Details

For many-to-one comparisons (pairwise comparisons with one control) in a two factorial unreplicated complete block design with non-normally distributed residuals, Nemenyi's test can be performed on Friedman-type ranked data.

Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . A total of  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : \theta_0 = \theta_i$  is tested in the two-tailed case against  $A_i : \theta_0 \neq \theta_i$ , ( $1 \leq i \leq m$ ).

The  $p$ -values are computed from the multivariate normal distribution.

### Value

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the  $p$ -values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Hollander M., Wolfe D. A., Chicken E. (2014), *Nonparametric Statistical Methods*. 3rd ed. New York: Wiley. 2014.

Miller Jr., R. G. (1996), *Simultaneous Statistical Inference*. New York: McGraw-Hill.

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

Siegel, S., Castellan Jr., N. J. (1988), *Nonparametric Statistics for the Behavioral Sciences*. 2nd ed. New York: McGraw-Hill.

Zarr J. H. (1999), *Biostatistical Analysis*. 4th ed. Upper Saddle River: Prentice-Hall.

## See Also

[friedmanTest](#), [friedman.test](#), [frdManyOneExactTest](#), [frdManyOneDemsarTest](#) [pmvnorm](#)

## 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.
## Assume A is the control.

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]))

## Global Friedman test
friedmanTest(y)

## Demsar's many-one test
frdManyOneDemsarTest(y=y, p.adjust = "bonferroni")

## Exact many-one test
frdManyOneExactTest(y=y, p.adjust = "bonferroni")

## Nemenyi's many-one test
frdManyOneNemenyiTest(y=y)
```

---

friedmanTest	<i>Friedman Rank Sum Test</i>
--------------	-------------------------------

---

### Description

Performs a Friedman rank sum test. The null hypothesis  $H_0 : \theta_i = \theta_j$  ( $i \neq j$ ) is tested against the alternative  $H_A : \theta_i \neq \theta_j$ , with at least one inequality being strict.

### Usage

```
friedmanTest(y, ...)

## Default S3 method:
friedmanTest(y, groups, blocks, dist = c("Chisquare",
    "FDist"), ...)
```

### Arguments

y	a numeric vector of data values, or a list of numeric data vectors.
groups	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
blocks	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
dist	the test distribution. Defaults to Chi square.
...	further arguments to be passed to or from methods.

### Details

The function has implemented Friedman's test as well as the extension of Conover and Iman (1981). Friedman's test statistic is asymptotically chi-squared distributed. Consequently, the default test distribution is `dist = "Chisquare"`.

If `dist = "FDist"` is selected, then the approach of Conover and Imam (1981) is performed. The Friedman Test using the  $F$ -distribution leads to the same results as doing an two-way Analysis of Variance without interaction on rank transformed data.

### Value

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

## References

W. J. Conover, R. L. Iman (1981), Rank transformations as a bridge between parametric and non-parametric statistics, *The American Statistician* 35, 124–129.

L. Sachs (1997), *Angewandte Statistik*. Berlin: Springer.

## See Also

[friedman.test](#)

## Examples

```
## Hollander & Wolfe (1973), p. 140ff.
## Comparison of three methods ("round out", "narrow angle", and
## "wide angle") for rounding first base. For each of 18 players
## and the three method, the average time of two runs from a point on
## the first base line 35ft from home plate to a point 15ft short of
## second base is recorded.
RoundingTimes <-
matrix(c(5.40, 5.50, 5.55,
        5.85, 5.70, 5.75,
        5.20, 5.60, 5.50,
        5.55, 5.50, 5.40,
        5.90, 5.85, 5.70,
        5.45, 5.55, 5.60,
        5.40, 5.40, 5.35,
        5.45, 5.50, 5.35,
        5.25, 5.15, 5.00,
        5.85, 5.80, 5.70,
        5.25, 5.20, 5.10,
        5.65, 5.55, 5.45,
        5.60, 5.35, 5.45,
        5.05, 5.00, 4.95,
        5.50, 5.50, 5.40,
        5.45, 5.55, 5.50,
        5.55, 5.55, 5.35,
        5.45, 5.50, 5.55,
        5.50, 5.45, 5.25,
        5.65, 5.60, 5.40,
        5.70, 5.65, 5.55,
        6.30, 6.30, 6.25),
      nrow = 22,
      byrow = TRUE,
      dimnames = list(1 : 22,
                      c("Round Out", "Narrow Angle", "Wide Angle")))

## Chisquare distribution
friedmanTest(RoundingTimes)
```

```
## check with friedman.test from R stats
friedman.test(RoundingTimes)

## F-distribution
friedmanTest(RoundingTimes, dist = "FDist")

## Check with One-way repeated measure ANOVA
rmat <- RoundingTimes
for (i in 1:length(RoundingTimes[,1])) rmat[i,] <- rank(rmat[i,])
dataf <- data.frame(
  y = y <- as.vector(rmat),
  g = g <- factor(c(col(RoundingTimes))),
  b = b <- factor(c(row(RoundingTimes))))
summary(aov(y ~ g + Error(b), data = dataf))
```

---

gamesHowellTest

*Games-Howell Test*

---

## Description

Performs Games-Howell all-pairs comparison test for normally distributed data with unequal group variances.

## Usage

```
gamesHowellTest(x, ...)

## Default S3 method:
gamesHowellTest(x, g, ...)

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

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .



**Details**

For all-pairs comparisons in an one-factorial layout with normally distributed residuals but unequal between-groups variances the Games-Howell Test can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

The p-values are computed from the studentized range distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**See Also**

[ptukey](#)

**Examples**

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1:5, each=5)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 != varN
anova(fit)
summary(dunnettT3Test(x, g))
```

**Description**

Performs Rosner's generalized extreme studentized deviate procedure to detect up-to `maxr` outliers in a univariate sample that follows an approximately normal distribution.

**Usage**

```
gesdTest(x, maxr)
```

**Arguments**

x                    a numeric vector of data.  
maxr                the maximum number of outliers to be tested.

**References**

Rosner, B. (1983), Percentage Points for a Generalized ESD Many-Outlier Procedure, *Technometrics*, 25, 165–172.

**Examples**

```
## Taken from Rosner (1983):
x <- c(-0.25,0.68,0.94,1.15,1.20,1.26,1.26,
1.34,1.38,1.43,1.49,1.49,1.55,1.56,
1.58,1.65,1.69,1.70,1.76,1.77,1.81,
1.91,1.94,1.96,1.99,2.06,2.09,2.10,
2.14,2.15,2.23,2.24,2.26,2.35,2.37,
2.40,2.47,2.54,2.62,2.64,2.90,2.92,
2.92,2.93,3.21,3.26,3.30,3.59,3.68,
4.30,4.64,5.34,5.42,6.01)

out <- gesdTest(x, 10)

## print method
out

## summary method
summary(out)
```

---

goreTest

*Gore Test*

---

**Description**

Performs Gore's test. The null hypothesis  $H_0 : \theta_i = \theta_j$  ( $i \neq j$ ) is tested against the alternative  $H_A : \theta_i \neq \theta_j$ , with at least one inequality being strict.

**Usage**

```
goreTest(y, groups, blocks)
```

**Arguments**

y	a numeric vector of data values.
groups	a vector or factor object giving the group for the corresponding elements of "y".
blocks	a vector or factor object giving the group for the corresponding elements of "y".

**Details**

The function has implemented Gore's test for testing main effects in unbalanced CRB designs, i.e. there are one or more observations per cell. The statistic is asymptotically chi-squared distributed.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

Gore, A. P. (1975) Some nonparametric tests and selection procedures for main effects in two-way layouts. *Ann. Inst. Stat. Math.* 27, 487–500.

**See Also**

[friedmanTest](#), [skillingsMackTest](#), [durbinTest](#)

**Examples**

```
## Crop Yield of 3 varieties on two
## soil classes
X <-c("130,A,Light
115,A,Light
123,A,Light
142,A,Light
117,A,Heavy
125,A,Heavy
139,A,Heavy
108,B,Light
114,B,Light
124,B,Light
106,B,Light
91,B,Heavy")
```

```

111,B,Heavy
110,B,Heavy
155,C,Light
146,C,Light
151,C,Light
165,C,Light
97,C,Heavy
108,C,Heavy")
con <- textConnection(X)
x <- read.table(con, header=FALSE, sep=",")
colnames(x) <- c("Yield", "Variety", "SoilType")
goreTest(y=x$Yield, groups=x$Variety, blocks=x$SoilType)

```

Grubbs

*Grubbs distribution***Description**

Distribution function and quantile function for Grubbs distribution.

**Usage**

```

qgrubbs(p, n)

pgrubbs(q, n, lower.tail = TRUE)

```

**Arguments**

p	vector of probabilities.
n	total sample size.
q	vector of quantiles.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

**Value**

pgrubbs gives the distribution function and qgrubbs gives the quantile function.

**References**

Grubbs, F.E. (1950) Sample criteria for testing outlying observations. *Ann. Math. Stat.* 21, 27–58.  
 Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.* <http://dx.doi.org/10.1007/s10182-011-0185-y>.

**See Also**

[TDist](#)

**Examples**

```
qgrubbs(0.05, 7)
```

grubbsTest

*Grubbs Outlier Test***Description**

Performs Grubbs single outlier test.

**Usage**

```
grubbsTest(x, alternative = c("two.sided", "greater", "less"))
```

**Arguments**

**x** a numeric vector of data.  
**alternative** the alternative hypothesis. Defaults to "two.sided".

**Details**

Let  $X$  denote an identically and independently distributed continuous variate with realizations  $x_i$  ( $1 \leq i \leq k$ ). Further, let the increasingly ordered realizations denote  $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$ . Then the following model for a single maximum outlier can be proposed:

$$x_{(i)} = \begin{cases} \mu + \epsilon_{(i)}, & i = 1, \dots, n-1 \\ \mu + \Delta + \epsilon_{(n)} & \end{cases}$$

with  $\epsilon \approx N(0, \sigma)$ . The null hypothesis,  $H_0 : \Delta = 0$  is tested against the alternative,  $H_A : \Delta > 0$ .

For testing a single minimum outlier, the model can be proposed as

$$x_{(i)} = \begin{cases} \mu + \Delta + \epsilon_{(1)} & i = 1 \\ \mu + \epsilon_{(i)}, & i = 2, \dots, n \end{cases}$$

The null hypothesis is tested against the alternative,  $H_A : \Delta < 0$ .

The p-value is computed with the function [pgrubbs](#).

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

## References

- Grubbs, F.E. (1950) Sample criteria for testing outlying observations. *Ann. Math. Stat.* 21, 27–58.
- Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.* <http://dx.doi.org/10.1007/s10182-011-0185-y>.

## Examples

```
data(Pentosan)
dat <- subset(Pentosan, subset = (material == "A"))
labMeans <- tapply(dat$value, dat$lab, mean)
grubbsTest(x = labMeans, alternative = "two.sided")
```

---

johnsonTest

*Testing against Ordered Alternatives (Johnson-Mehrotra Test)*

---

## Description

Performs the Johnson-Mehrotra test for testing against ordered alternatives in a balanced one-factorial sampling design.

## Usage

```
johnsonTest(x, ...)

## Default S3 method:
johnsonTest(x, g, alternative = c("two.sided", "greater",
  "less"), ...)

## S3 method for class 'formula'
johnsonTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), ...)
```

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "two.sided".
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 <a href="#">model.frame</a> ) 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").

**Details**

The null hypothesis,  $H_0 : \theta_1 = \theta_2 = \dots = \theta_k$  is tested against a simple order hypothesis,  $H_A : \theta_1 \leq \theta_2 \leq \dots \leq \theta_k, \theta_1 < \theta_k$ .

The p-values are estimated from the standard normal distribution.

**Value**

A list with class "hstest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

Bortz, J. (1993). *Statistik für Sozialwissenschaftler* (4th ed.). Berlin: Springer.

Johnson, R. A., & Mehrotra, K. G. (1972). Some c-sample nonparametric tests for ordered alternatives. *Journal of the Indian Statistical Association*, 9, 8–23.

**See Also**

[kruskalTest](#) and [shirleyWilliamsTest](#) of the package **PMCMRplus**, [kruskal.test](#) of the library **stats**.

**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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
```

```

jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)

```

---

jonckheereTest            *Testing against Ordered Alternatives (Jonckheere-Terpstrata Test)*

---

### Description

Performs the Jonckheere-Terpstrata test for testing against ordered alternatives.

### Usage

```

jonckheereTest(x, ...)

## Default S3 method:
jonckheereTest(x, g, alternative = c("two.sided", "greater",
  "less"), continuity = FALSE, ...)

## S3 method for class 'formula'
jonckheereTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), continuity = FALSE, ...)

```

### Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "two.sided".
continuity	logical indicator whether a continuity correction shall be performed. Defaults to FALSE.
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 <a href="#">model.frame</a> ) 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").



**Details**

The null hypothesis,  $H_0 : \theta_1 = \theta_2 = \dots = \theta_k$  is tested against a simple order hypothesis,  $H_A : \theta_1 \leq \theta_2 \leq \dots \leq \theta_k, \theta_1 < \theta_k$ .

The p-values are estimated from the standard normal distribution.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**Source**

The code for the computation of the standard deviation for the Jonckheere-Terpstrata test in the presence of ties was taken from:

Kloke, J., McKean, J. (2016). npsm: Package for Nonparametric Statistical Methods using R. R package version 0.5. <https://CRAN.R-project.org/package=npsm>

**Note**

`jonckheere.test(x, g, alternative = "two.sided", continuity = TRUE)` is equivalent to `cor.test(x, as.numeric(g), method = "kendall", alternative = "two.sided", continuity = TRUE)`

**References**

Jonckheere, A. R. (1954). A distribution-free k-sample test against ordered alternatives. *Biometrika*, 41, 133–145.

Kloke, J., McKean, J. W. (2015). *Nonparametric statistical methods using R*. Boca Raton, FL: Chapman & Hall/CRC.

**See Also**

`kruskalTest` and `shirleyWilliamsTest` of the package **PMCMRplus**, `kruskal.test` of the library **stats**.

**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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)
```

---

kruskalTest

*Kruskal-Wallis Rank Sum Test*

---

**Description**

Performs a Kruskal-Wallis rank sum test.

**Usage**

```
kruskalTest(x, ...)
```

## Default S3 method:

```
kruskalTest(x, g, dist = c("Chisquare", "KruskalWallis",
  "FDist"), ...)
```

## S3 method for class 'formula'

```
kruskalTest(formula, data, subset, na.action,
  dist = c("Chisquare", "KruskalWallis", "FDist"), ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
dist	the test distribution. Defaults's to "Chisquare".
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 <code>model.frame</code> ) 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> .

**Details**

For one-factorial designs with non-normally distributed residuals the Kruskal-Wallis rank sum test can be performed to test the  $H_0 : F_1(x) = F_2(x) = \dots = F_k(x)$  against the  $H_A : F_i(x) \neq F_j(x) (i \neq j)$  with at least one strict inequality.

As the Kruskal-Wallis H-statistic is asymptotically chi-squared distributed with  $v = k - 1$  degree of freedom, the default test distribution is consequently `dist = "Chisquare"`. If `dist = "KruskalWallis"` is selected, an incomplete beta approximation is used for the calculation of p-values as implemented in the function `pKruskalWallis` of the package **SuppDists**. For `dist = "FDist"` the proposed method of Conover and Imam (1981) is used, which is equivalent to a one-way ANOVA F-test using rank transformed data (see examples).

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

W. J. Conover, R. L. Iman (1981) Rank transformations as a bridge between parametric and non-parametric statistics, *The American Statistician* 35, 124–129.

L. Sachs (1997), *Angewandte Statistik*. Berlin: Springer.

**See Also**

[kruskal.test](#), [pKruskalWallis](#), [Chisquare](#), [FDist](#)

**Examples**

```
## Hollander & Wolfe (1973), 116.
## Mucociliary efficiency from the rate of removal of dust in normal
## subjects, subjects with obstructive airway disease, and subjects
## with asbestosis.
x <- c(2.9, 3.0, 2.5, 2.6, 3.2) # normal subjects
y <- c(3.8, 2.7, 4.0, 2.4)     # with obstructive airway disease
z <- c(2.8, 3.4, 3.7, 2.2, 2.0) # with asbestosis

datf <- data.frame(
  g = g <- c(rep("ns", length(x)), rep("oad",
    length(y)), rep("a", length(z))),
  x = x <- c(x, y, z))

## Using incomplete beta approximation
kruskalTest(x ~ g, datf, dist="KruskalWallis")

## Using chisquare distribution
kruskalTest(x ~ g, datf, dist="Chisquare")

## Check with kruskal.test from R stats
kruskal.test(x ~ g, datf)

## Using Conover's F
kruskalTest(x ~ g, datf, dist="FDist")

## Check with aov on ranks
anova(aov(rank(x) ~ g, datf))

## Check with oneway.test
oneway.test(rank(x) ~ g, datf, var.equal=FALSE)
```

---

kwAllPairsConoverTest *Conover's All-Pairs Rank Comparison Test*

---

**Description**

Performs Conover's non-parametric all-pairs comparison test for Kruskal-type ranked data.

**Usage**

```
kwAllPairsConoverTest(x, ...)

## Default S3 method:
kwAllPairsConoverTest(x, g,
```

```

p.adjust.method = c("single-step", p.adjust.methods), ...)

## S3 method for class 'formula'
kwAllPairsConoverTest(formula, data, subset, na.action,
  p.adjust.method = c("single-step", p.adjust.methods), ...)

```

### Arguments

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

### Details

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Conover's non-parametric test can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

If `p.adjust.method == "single-step"` the p-values are computed from the studentized range distribution. Otherwise, the p-values are computed from the t-distribution using any of the p-adjustment methods as included in [p.adjust](#).

### Value

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for p-value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

**References**

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

**See Also**

[Tukey](#), [TDist](#), [p.adjust](#), [kruskalTest](#), [kwAllPairsDunnTest](#), [kwAllPairsNemenyiTest](#)

**Examples**

```
## Data set InsectSprays
## Global test
kruskalTest(count ~ spray, data = InsectSprays)

## Conover's all-pairs comparison test
## single-step means Tukey's p-adjustment
ans <- kwAllPairsConoverTest(count ~ spray, data = InsectSprays,
                             p.adjust.method = "single-step")
summary(ans)

## Dunn's all-pairs comparison test
ans <- kwAllPairsDunnTest(count ~ spray, data = InsectSprays,
                          p.adjust.method = "bonferroni")
summary(ans)

## Nemenyi's all-pairs comparison test
ans <- kwAllPairsNemenyiTest(count ~ spray, data = InsectSprays)
summary(ans)
```

---

kwAllPairsDunnTest      *Dunn's All-Pairs Rank Comparison Test*

---

**Description**

Performs Dunn's non-parametric all-pairs comparison test for Kruskal-type ranked data.

**Usage**

```
kwAllPairsDunnTest(x, ...)

## Default S3 method:
kwAllPairsDunnTest(x, g, p.adjust.method = p.adjust.methods,
                  ...)

## S3 method for class 'formula'
kwAllPairsDunnTest(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.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Dunn's non-parametric test can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x), i \neq j$ .

The p-values are computed from the standard normal distribution using any of the p-adjustment methods as included in [p.adjust](#). Originally, Dunn (1964) proposed Bonferroni's p-adjustment method.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

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

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

**See Also**

[Normal](#), [p.adjust](#), [kruskalTest](#), [kwAllPairsConoverTest](#), [kwAllPairsNemenyiTest](#)

**Examples**

```
## Data set InsectSprays
## Global test
kruskalTest(count ~ spray, data = InsectSprays)

## Conover's all-pairs comparison test
## single-step means Tukey's p-adjustment
ans <- kwAllPairsConoverTest(count ~ spray, data = InsectSprays,
                             p.adjust.method = "single-step")
summary(ans)

## Dunn's all-pairs comparison test
ans <- kwAllPairsDunnTest(count ~ spray, data = InsectSprays,
                          p.adjust.method = "bonferroni")
summary(ans)

## Nemenyi's all-pairs comparison test
ans <- kwAllPairsNemenyiTest(count ~ spray, data = InsectSprays)
summary(ans)
```

---

kwAllPairsNemenyiTest *Nemenyi's All-Pairs Rank Comparison Test*

---

**Description**

Performs Nemenyi's non-parametric all-pairs comparison test for Kruskal-type ranked data.

**Usage**

```
kwAllPairsNemenyiTest(x, ...)

## Default S3 method:
kwAllPairsNemenyiTest(x, g, dist = c("Tukey", "Chisquare"),
                      ...)

## S3 method for class 'formula'
kwAllPairsNemenyiTest(formula, data, subset, na.action,
                      dist = c("Tukey", "Chisquare"), ...)
```

**Arguments**

x                    a numeric vector of data values, or a list of numeric data vectors.  
 ...                 further arguments to be passed to or from methods.



<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>dist</code>	the distribution for determining the p-value. Defaults to "Tukey".
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

### Details

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Nemenyi's non-parametric test can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

If `dist == "Tukey"` is selected, the p-values are computed from the studentized range distribution. If `dist == "Chisquare"` is selected, the p-values are computed from the chi-square distribution.

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

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

L. Sachs (1997), *Angewandte Statistik*. Berlin: Springer.

F. Wilcoxon, R. A. Wilcoxon (1964), *Some rapid approximate statistical procedures*. Pearl River: Lederle Laboratories.

### See Also

[Tukey](#), [Chisquare](#), [p.adjust](#), [kruskalTest](#), [kwAllPairsDunnTest](#), [kwAllPairsConoverTest](#)

**Examples**

```
## Data set InsectSprays
## Global test
kruskalTest(count ~ spray, data = InsectSprays)

## Conover's all-pairs comparison test
## single-step means Tukey's p-adjustment
ans <- kwAllPairsConoverTest(count ~ spray, data = InsectSprays,
                             p.adjust.method = "single-step")
summary(ans)

## Dunn's all-pairs comparison test
ans <- kwAllPairsDunnTest(count ~ spray, data = InsectSprays,
                           p.adjust.method = "bonferroni")
summary(ans)

## Nemenyi's all-pairs comparison test
ans <- kwAllPairsNemenyiTest(count ~ spray, data = InsectSprays)
summary(ans)
```

---

kwManyOneConoverTest *Conover's Many-to-One Rank Comparison Test*

---

**Description**

Performs Conover's non-parametric many-to-one comparison test for Kruskal-type ranked data.

**Usage**

```
kwManyOneConoverTest(x, ...)

## Default S3 method:
kwManyOneConoverTest(x, g, alternative = c("two.sided",
    "greater", "less"), p.adjust.method = c("single-step", p.adjust.methods),
    ...)

## S3 method for class 'formula'
kwManyOneConoverTest(formula, data, subset, na.action,
    alternative = c("two.sided", "greater", "less"),
    p.adjust.method = c("single-step", p.adjust.methods), ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.

alternative	the alternative hypothesis. Defaults to two.sided.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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").

### Details

For many-to-one comparisons (pairwise comparisons with one control) in an one-factorial layout with non-normally distributed residuals Conover's non-parametric test can be performed. Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . Then  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : \theta_0 = \theta_i$  is tested in the two-tailed case against  $A_i : \theta_0 \neq \theta_i$ , ( $1 \leq i \leq m$ ).

If `p.adjust.method == "single-step"` is selected, the  $p$ -values will be computed from the multivariate  $t$  distribution. Otherwise, the  $p$ -values are computed from the  $t$ -distribution using any of the  $p$ -adjustment methods as included in [p.adjust](#).

### Value

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for p-value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

### References

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

### See Also

[pmvt](#), [TDist](#), [kruskalTest](#), [kwManyOneDunnTest](#), [kwManyOneNdwTest](#)

**Examples**

```
## Data set PlantGrowth
## Global test
kruskalTest(weight ~ group, data = PlantGrowth)

## Conover's many-one comparison test
## single-step means p-value from multivariate t distribution
ans <- kwManyOneConoverTest(weight ~ group, data = PlantGrowth,
                             p.adjust.method = "single-step")
summary(ans)

## Conover's many-one comparison test
ans <- kwManyOneConoverTest(weight ~ group, data = PlantGrowth,
                             p.adjust.method = "holm")
summary(ans)

## Dunn's many-one comparison test
ans <- kwManyOneDunnTest(weight ~ group, data = PlantGrowth,
                          p.adjust.method = "holm")
summary(ans)

## Nemenyi's many-one comparison test
ans <- kwManyOneNdwTest(weight ~ group, data = PlantGrowth,
                         p.adjust.method = "holm")
summary(ans)
```

---

kwManyOneDunnTest	<i>Dunn's Many-to-One Rank Comparison Test</i>
-------------------	--

---

**Description**

Performs Dunn's non-parametric many-to-one comparison test for Kruskal-type ranked data.

**Usage**

```
kwManyOneDunnTest(x, ...)

## Default S3 method:
kwManyOneDunnTest(x, g, alternative = c("two.sided",
    "greater", "less"), p.adjust.method = c("single-step", p.adjust.methods),
    ...)

## S3 method for class 'formula'
kwManyOneDunnTest(formula, data, subset, na.action,
    alternative = c("two.sided", "greater", "less"),
    p.adjust.method = c("single-step", p.adjust.methods), ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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").

**Details**

For many-to-one comparisons (pairwise comparisons with one control) in an one-factorial layout with non-normally distributed residuals Dunn's non-parametric test can be performed. Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . Then  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : \theta_0 = \theta_i$  is tested in the two-tailed case against  $A_i : \theta_0 \neq \theta_i$ , ( $1 \leq i \leq m$ ).

If `p.adjust.method == "single-step"` is selected, the  $p$ -values will be computed from the multivariate normal distribution. Otherwise, the  $p$ -values are computed from the standard normal distribution using any of the  $p$ -adjustment methods as included in [p.adjust](#).

**Value**

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for p-value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

**References**

- O. J. Dunn (1964). Multiple comparisons using rank sums. *Technometrics*, 6, 241-252.
- S. Siegel, N. J. Castellan Jr. (1988), *Nonparametric Statistics for The Behavioral Sciences*. New York: McGraw-Hill.

**See Also**

[pmvnorm](#), [TDist](#), [kruskalTest](#), [kwManyOneConoverTest](#), [kwManyOneNdwTest](#)

**Examples**

```
## Data set PlantGrowth
## Global test
kruskalTest(weight ~ group, data = PlantGrowth)

## Conover's many-one comparison test
## single-step means p-value from multivariate t distribution
ans <- kwManyOneConoverTest(weight ~ group, data = PlantGrowth,
                             p.adjust.method = "single-step")
summary(ans)

## Conover's many-one comparison test
ans <- kwManyOneConoverTest(weight ~ group, data = PlantGrowth,
                             p.adjust.method = "holm")
summary(ans)

## Dunn's many-one comparison test
ans <- kwManyOneDunnTest(weight ~ group, data = PlantGrowth,
                          p.adjust.method = "holm")
summary(ans)

## Nemenyi's many-one comparison test
ans <- kwManyOneNdwTest(weight ~ group, data = PlantGrowth,
                        p.adjust.method = "holm")
summary(ans)
```

---

kwManyOneNdwTest

*Nemenyi-Damico-Wolfe Many-to-One Rank Comparison Test*


---

**Description**

Performs Nemenyi-Damico-Wolfe non-parametric many-to-one comparison test for Kruskal-type ranked data.

**Usage**

```
kwManyOneNdwTest(x, ...)

## Default S3 method:
kwManyOneNdwTest(x, g, alternative = c("two.sided",
    "greater", "less"), p.adjust.method = c("single-step", p.adjust.methods),
    ...)

## S3 method for class 'formula'
kwManyOneNdwTest(formula, data, subset, na.action,
```

```
alternative = c("two.sided", "greater", "less"),
p.adjust.method = c("single-step", p.adjust.methods), ...)
```

### Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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").

### Details

For many-to-one comparisons (pairwise comparisons with one control) in an one-factorial layout with non-normally distributed residuals the Nemenyi-Damico-Wolfe non-parametric test can be performed. Let there be  $k$  groups including the control, then the number of treatment levels is  $m = k - 1$ . Then  $m$  pairwise comparisons can be performed between the  $i$ -th treatment level and the control.  $H_i : \theta_0 = \theta_i$  is tested in the two-tailed case against  $A_i : \theta_0 \neq \theta_i$ , ( $1 \leq i \leq m$ ).

If `p.adjust.method == "single-step"` is selected, the  $p$ -values will be computed from the multivariate normal distribution. Otherwise, the  $p$ -values are computed from the standard normal distribution using any of the  $p$ -adjustment methods as included in [p.adjust](#).

### Value

A list with class "PMCMR" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- p.value** lower-triangle matrix of the p-values for the pairwise tests.
- alternative** a character string describing the alternative hypothesis.
- p.adjust.method** a character string describing the method for p-value adjustment.
- model** a data frame of the input data.
- dist** a string that denotes the test distribution.

**Note**

This function is essentially the same as [kwManyOneDunnTest](#), but there is no tie correction included. Therefore, the implementation of Dunn's test is superior, when ties are present.

**References**

Damico, J. A., Wolfe, D. A. (1989), Extended tables of the exact distribution of a rank statistic for treatments versus control multiple comparisons in one-way layout designs. *Communications in Statistics - Theory and Methods*, 18(9), 3327–3353.

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

**See Also**

[pmt](#), [TDist](#), [kruskalTest](#), [kwManyOneDunnTest](#), [kwManyOneConoverTest](#)

**Examples**

```
## Data set PlantGrowth
## Global test
kruskalTest(weight ~ group, data = PlantGrowth)

## Conover's many-one comparison test
## single-step means p-value from multivariate t distribution
ans <- kwManyOneConoverTest(weight ~ group, data = PlantGrowth,
                             p.adjust.method = "single-step")
summary(ans)

## Conover's many-one comparison test
ans <- kwManyOneConoverTest(weight ~ group, data = PlantGrowth,
                             p.adjust.method = "holm")
summary(ans)

## Dunn's many-one comparison test
ans <- kwManyOneDunnTest(weight ~ group, data = PlantGrowth,
                          p.adjust.method = "holm")
summary(ans)

## Nemenyi's many-one comparison test
ans <- kwManyOneNdwTest(weight ~ group, data = PlantGrowth,
                         p.adjust.method = "holm")
summary(ans)
```

**Description**

Performs Le's test for testing against ordered alternatives.



**Usage**

```
leTest(x, ...)

## Default S3 method:
leTest(x, g, alternative = c("two.sided", "greater",
  "less"), ...)

## S3 method for class 'formula'
leTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "two.sided".
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 <a href="#">model.frame</a> ) 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").

**Details**

The null hypothesis,  $H_0 : \theta_1 = \theta_2 = \dots = \theta_k$  is tested against a simple order hypothesis,  $H_A : \theta_1 \leq \theta_2 \leq \dots \leq \theta_k, \theta_1 < \theta_k$ .

The p-values are estimated from the standard normal distribution.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

## References

Le, C. T. (1988). A new rank test against ordered alternatives in k-sample problems. *Biometrical Journal*, 30, 87–92.

## See Also

[kruskalTest](#) and [shirleyWilliamsTest](#) of the package **PMCMRplus**, [kruskal.test](#) of the library **stats**.

## 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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)
```

---

lsdTest

*Least Significant Difference Test*

---

## Description

Performs the least significant difference all-pairs comparisons test for normally distributed data with equal group variances.

**Usage**

```

lsdTest(x, ...)

## Default S3 method:
lsdTest(x, g, ...)

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

```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with normally distributed residuals and equal variances the least significant difference test can be performed. A total of  $m = k(k-1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

The p-values are computed from the t-distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**Note**

As there is no p-value adjustment build in, this function is equivalent to Fisher's protected LSD test, provided that the LSD test is only applied after a significant one-way ANOVA F-test. If one is interested in other types of LSD test (i.e. with p-value adjustment) see function [pairwise.t.test](#).

**References**

L. Sachs (1997) *Angewandte Statistik*, New York: Springer.

**See Also**

[TDist](#), [pairwise.t.test](#)

**Examples**

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1, 25)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 = varN
anova(fit)
summary(lsdTest(x, g))
```

---

mackWolfeTest

*Mack-Wolfe Test for Umbrella Alternatives*

---

**Description**

Performs Mack-Wolfe non-parametric test for umbrella alternatives.

**Usage**

```
mackWolfeTest(x, ...)

## Default S3 method:
mackWolfeTest(x, g, p = NULL, nperm = 1000, ...)

## S3 method for class 'formula'
mackWolfeTest(formula, data, subset, na.action, p = NULL,
              nperm = 1000, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
p	the a-priori known peak as an ordinal number of the treatment group including the zero dose level, i.e. $p = \{1, \dots, k\}$ . Defaults to NULL.
nperm	number of permutations for the asymptotic permutation test. Defaults to 1000.
formula	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
data	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> .
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> .

**Details**

In dose-finding studies one may assume an increasing treatment effect with increasing dose level. However, the test subject may actually succumb to toxic effects at high doses, which leads to decreasing treatment effects.

The scope of the Mack-Wolfe Test is to test for umbrella alternatives for either a known or unknown point  $p$  (i.e. dose-level), where the peak (umbrella point) is present.

$H_i : \theta_0 = \theta_i = \dots = \theta_k$  is tested against the alternative  $A_i : \theta_1 \leq \dots \theta_p \geq \theta_k$  for some  $p$ , with at least one strict inequality.

If `p = NULL` (peak unknown), the upper-tail  $p$ -value is computed via an asymptotic bootstrap permutation test.

If an integer value for `p` is given (peak known), the upper-tail  $p$ -value is computed from the standard normal distribution (`pnorm`).

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the  $p$ -value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**Note**

One may increase the number of permutations to e.g. `nperm = 10000` in order to get more precise p-values. However, this will be on the expense of computational time.

**References**

- Chen, I.Y. (1991) Notes on the Mack-Wolfe and Chen-Wolfe Tests for Umbrella Alternatives. *Biom. J.*, 33, 281–290.
- Mack, G.A., Wolfe, D. A. (1981) K-sample rank tests for umbrella alternatives. *J. Amer. Statist. Assoc.*, 76, 175–181.

**See Also**

[pnorm](#), [sample](#).

**Examples**

```
## Example from Table 6.10 of Hollander and Wolfe (1999).
## Plates with Salmonella bacteria of strain TA98 were exposed to
## various doses of Acid Red 114 (in mu g / ml).
## The data are the numbers of visible revertant colonies on 12 plates.
## Assume a peak at D333 (i.e. p = 3).
x <- c(22, 23, 35, 60, 59, 54, 98, 78, 50, 60, 82, 59, 22, 44,
      33, 23, 21, 25)
g <- as.ordered(rep(c(0, 100, 333, 1000, 3333, 10000), each=3))
plot(x ~ g)
mackWolfeTest(x=x, g=g, p=3)
```

---

Mandel-h

*Mandel's h distribution*


---

**Description**

Distribution function and quantile function for Mandel's h distribution.

**Usage**

```
qmandelh(p, k, lower.tail = TRUE, log.p = FALSE)
```

```
pmandelh(q, k, lower.tail = TRUE, log.p = FALSE)
```

**Arguments**

<code>p</code>	vector of probabilities.
<code>k</code>	number of groups.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .
<code>log.p</code>	logical; if TRUE, probabilities are given as $\log(p)$ .
<code>q</code>	vector of quantiles.

**Value**

`pmandelk` gives the distribution function and `qmandelk` gives the quantile function.

**Source**

The code for `pmandelk` was taken from:  
 Stephen L R Ellison. (2017). `metRology`: Support for Metrological Applications. R package version 0.9-26-2. <https://CRAN.R-project.org/package=metRology>

**References**

Practice E 691, 2005, *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

**See Also**

[mandelkTest](#)

**Examples**

```
## We need a two-sided upper-tail quantile
qmandelk(p = 0.005/2, k = 7, lower.tail=FALSE)
```

---

Mandel-k

*Mandel's k distribution*


---

**Description**

Distribution function and quantile function for Mandel's k distribution.

**Usage**

```
qmandelk(p, k, n, lower.tail = TRUE, log.p = FALSE)
```

```
pmandelk(q, k, n, lower.tail = TRUE, log.p = FALSE)
```

**Arguments**

<code>p</code>	vector of probabilities.
<code>k</code>	number of groups.
<code>n</code>	number of replicates per group.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .
<code>log.p</code>	logical; if TRUE, probabilities are given as $\log(p)$ .
<code>q</code>	vector of quantiles.

**Value**

pmandelk gives the distribution function and qmandelk gives the quantile function.

**Source**

The code for pmandelk was taken from:

Stephen L R Ellison. (2017). metRology: Support for Metrological Applications. R package version 0.9-26-2. <https://CRAN.R-project.org/package=metRology>

**Note**

The functions are only appropriate for balanced designs.

**References**

Practice E 691, 2005, *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

**See Also**

[mandelkTest](#)

[pmandelh](#), [qmandelh](#)

**Examples**

```
qmandelk(0.005, 7, 3, lower.tail=FALSE)
```

---

mandelhTest

*Mandel's h test according to E 651 ASTM*

---

**Description**

The function calculates the consistency statistics h and corresponding p-values for each group (lab) according to Practice E 691 ASTM.

**Usage**

```
mandelhTest(x, ...)
```

```
## Default S3 method:  
mandelhTest(x, g, ...)
```

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



### Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .

### Value

A list with class "mandel" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**p.value** the p-value for the test.

**statistic** the estimated quantiles of Mandel's statistic.

**alternative** a character string describing the alternative hypothesis.

**grouplev** a character vector describing the levels of the groups.

**nrofrepl** the number of replicates for each group.

### References

Practice E 691, 2005, *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

### See Also

[qmandelh](#) [pmandelh](#)

### Examples

```
data(Pentosan)
mandelhTest(value ~ lab, data=Pentosan, subset=(material == "A"))
```

---

mandelkTest

*Mandel's k test according to E 651 ASTM*


---

### Description

The function calculates the consistency statistics  $k$  and corresponding  $p$ -values for each group (lab) according to Practice E 691 ASTM.

### Usage

```
mandelkTest(x, ...)
```

```
## Default S3 method:
mandelkTest(x, g, ...)
```

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

### Arguments

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

### Value

A list with class "mandel" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**p.value** the  $p$ -value for the test.

**statistic** the estimated quantiles of Mandel's statistic.

**alternative** a character string describing the alternative hypothesis.

**grouplev** a character vector describing the levels of the groups.

**nrofrepl** the number of replicates for each group.

**References**

Practice E 691, 2005, *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

**See Also**

[qmandelk](#) [pmandelk](#)

**Examples**

```
data(Pentosan)
mandelkTest(value ~ lab, data=Pentosan, subset=(material == "A"))
```

---

manyOneUTest	<i>Multiple Comparisons with One Control (U-test)</i>
--------------	---

---

**Description**

Performs pairwise comparisons of multiple group levels with one control.

**Usage**

```
manyOneUTest(x, ...)

## Default S3 method:
manyOneUTest(x, g, alternative = c("two.sided", "greater",
  "less"), p.adjust.method = c("single-step", p.adjust.methods), ...)

## S3 method for class 'formula'
manyOneUTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = c("single-step", p.adjust.methods), ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> )
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 <a href="#">model.frame</a> ) 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> .

## Details

This functions performs Wilcoxon, Mann and Whitney's U-test for a one factorial design where each factor level is tested against one control ( $m = k - 1$  tests). As the data are re-ranked for each comparison, this test is only suitable for balanced (or almost balanced) experimental designs.

For the two-tailed test and `p.adjust.method = "single-step"` the multivariate normal distribution is used for controlling Type 1 error and to calculate p-values. Otherwise, the p-values are calculated from the standard normal distribution with any latter p-adjustment as available by [p.adjust](#).

## Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

OECD (ed. 2006), *Current approaches in the statistical analysis of ecotoxicity data: A guidance to application*. OECD Series on testing and assessment, No. 54.

## See Also

[wilcox.test](#), [pmvnorm](#), [Normal](#)

MTest

*Extended One-Sided Studentised Range Test***Description**

Performs Nashimoto-Wright's extended one-sided studentised range test against an ordered alternative for normal data with equal variances.

This test is an extension of Hayter's OSRT (see [osrtTest](#) by applying a simple order restriction of  $\mu_{m'} - \mu_m \leq \mu_j - \mu_i \leq \mu_{l'} - \mu_l$  for any  $l \leq i \leq m$  and  $m' \leq j \leq l'$ . It tests all-pairs  $H_{ij} : \mu_i \geq \mu_j$  against  $A_{ij} : \mu_i < \mu_j$  for any  $1 \leq i < j \leq k$ .

**Usage**

```
MTest(x, ...)

## Default S3 method:
MTest(x, g, ...)

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

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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").

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Nashimoto, K., Wright, F.T. (2005), Multiple comparison procedures for detecting differences in simply ordered means. *Comput. Statist. Data Anal.* 48, 291–306.

## Examples

```
MTest(weight ~ group, data = PlantGrowth)
```

---

```
normalScoresAllPairsTest
```

```
Lu-Smith All-Pairs Comparison Normal Scores Test
```

---

## Description

Performs Lu-Smith all-pairs comparison normal scores test.

## Usage

```
normalScoresAllPairsTest(x, ...)

## Default S3 method:
normalScoresAllPairsTest(x, g,
  p.adjust.method = c("single-step", p.adjust.methods), ...)

## S3 method for class 'formula'
normalScoresAllPairsTest(formula, data, subset, na.action,
  p.adjust.method = c("single-step", p.adjust.methods), ...)
```

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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> .

### Details

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Lu and Smith's normal scores transformation can be used prior to an all-pairs comparison test. A total of  $m = k(k-1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : F_i(x) = F_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : F_i(x) \neq F_j(x)$ ,  $i \neq j$ . For `p.adjust.method = "single-step"` the Tukey's studentized range distribution is used to calculate p-values (see `\codeTukey`). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by `p.adjust`.

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

Lu, H., Smith, P. (1979). Distribution of normal scores statistic for nonparametric one-way analysis of variance. *Journal of the American Statistical Association*, 74, 715–722.

### See Also

[normalScoresTest](#), [normalScoresManyOneTest](#), [normOrder](#).

---

 normalScoresManyOneTest

*Lu-Smith Many-One Comparisons Normal Scores Test*


---

## Description

Performs Lu-Smith multiple comparison normal scores test with one control.

## Usage

```
normalScoresManyOneTest(x, ...)

## Default S3 method:
normalScoresManyOneTest(x, g, alternative = c("two.sided",
  "greater", "less"), p.adjust.method = c("single-step", p.adjust.methods),
  ...)

## S3 method for class 'formula'
normalScoresManyOneTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = c("single-step", p.adjust.methods), ...)
```

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
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 <a href="#">model.frame</a> ) 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").

## Details

For many-to-one comparisons in an one-factorial layout with non-normally distributed residuals Lu and Smith's normal scores transformation can be used prior to a many-to-one comparison test. A total of  $m = k - 1$  hypotheses can be tested. The null hypothesis  $H_i : F_0(x) = F_i(x)$  is



tested in the two-tailed test against the alternative  $A_i : F_0(x) \neq F_i(x)$ ,  $1 \leq i \leq k - 1$ . For `p.adjust.method = "single-step"` the multivariate t distribution is used to calculate p-values (see `pmvt`). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by `p.adjust`.

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

Lu, H., Smith, P. (1979). Distribution of normal scores statistic for nonparametric one-way analysis of variance. *Journal of the American Statistical Association*, 74, 715–722.

### See Also

[normalScoresTest](#), [normalScoresAllPairsTest](#), [normOrder](#), [pmvt](#).

### Examples

```
## Data set PlantGrowth
## Global test
normalScoresTest(weight ~ group, data = PlantGrowth)

## Lu-Smith's many-one comparison test
ans <- normalScoresManyOneTest(weight ~ group, data = PlantGrowth, p.adjust.method = "holm")
summary(ans)
```

---

normalScoresTest

*Lu-Smith Normal Scores Test*

---

### Description

Performs the Lu-Smith normal score test

**Usage**

```
normalScoresTest(x, ...)

## Default S3 method:
normalScoresTest(x, g, ...)

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

**Arguments**

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<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> .

**Details**

For one-factorial designs with non-normally distributed residuals the Lu-Smith normal score test can be performed to test the  $H_0 : F_1(x) = F_2(x) = \dots = F_k(x)$  against the  $H_A : F_i(x) \neq F_j(x)$  ( $i \neq j$ ) with at least one strict inequality. This function is basically a wrapper function to `pNormScore` of the package **SuppDists**.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

**References**

Lu, H., Smith, P. (1979). Distribution of normal scores statistic for nonparametric one-way analysis of variance. *Journal of the American Statistical Association*, 74, 715–722.

**See Also**

[vanWaerdenTest](#), [kruskalTest](#), [pNormScore](#)

**Examples**

```
normalScoresTest(count ~ spray, data = InsectSprays)
```

---

NPMTest

*All-Pairs Comparisons for Simply Ordered Mean Ranksums*

---

**Description**

Performs Nashimoto and Wright's all-pairs comparison procedure for simply ordered mean ranksums. Their test denoted as NPM test is basically an extension of Nemenyi's procedure for testing increasingly ordered alternatives.

The modified procedure uses the property of a simple order,  $\theta'_m - \theta_m \leq \theta_j - \theta_i \leq \theta'_l - \theta_l$  ( $l \leq i \leq m$  and  $m' \leq j \leq l'$ ). The null hypothesis  $H_{ij} : \theta_i = \theta_j$  is tested against the alternative  $A_{ij} : \theta_i < \theta_j$  for any  $1 \leq i < j \leq k$ .

The p-values are estimated from the studentized range distribution. If the medians are already increasingly ordered, than the NPM-test simplifies to the ordinary Nemenyi test (see [kwAllPairsNemenyiTest](#)).

**Usage**

```
NPMTest(x, ...)

## Default S3 method:
NPMTest(x, g, ...)

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

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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.

<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

## Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Nashimoto, K., Wright, F.T., (2005), Multiple comparison procedures for detecting differences in simply ordered means. *Comput. Statist. Data Anal.* 48, 291–306.

## See Also

[kwAllPairsNemenyiTest](#)

## 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 <- gl(3,5)
levels(g) <- c("A", "B", "C")
NPMTest(x, g)
```

---

osrtTest	<i>One-Sided Studentised Range Test</i>
----------	---

---

### Description

Performs Hayter's one-sided studentised range test against an ordered alternative for normal data with equal variances.

### Usage

```
osrtTest(x, ...)

## Default S3 method:
osrtTest(x, g, ...)

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

### Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .

### Value

A list with class "htest" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** the estimated quantile of the test statistic.
- p.value** the p-value for the test.
- parameter** the parameters of the test statistic, if any.
- alternative** a character string describing the alternative hypothesis.
- estimates** the estimates, if any.
- null.value** the estimate under the null hypothesis, if any.

## References

A. J. Hayter (1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative. *Journal of the American Statistical Association* 85, 778–785.

## Examples

```
osrtTest(weight ~ group, data = PlantGrowth)
```

---

pageTest

*Page Rank Sum Test*

---

## Description

Performs Page's ordered aligned rank sum test.

## Usage

```
pageTest(y, ...)
```

```
## Default S3 method:
```

```
pageTest(y, groups, blocks, alternative = c("two.sided",
      "greater", "less"), ...)
```

## Arguments

y	a numeric vector of data values, or a list of numeric data vectors.
groups	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
blocks	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.
...	further arguments to be passed to or from methods.

## Value

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** the estimated quantile of the test statistic.

**p.value** the p-value for the test.

**parameter** the parameters of the test statistic, if any.

**alternative** a character string describing the alternative hypothesis.

**estimates** the estimates, if any.

**null.value** the estimate under the null hypothesis, if any.

## References

E. B. Page (1963), Ordered hypotheses for multiple treatments: A significance test for linear ranks. *Journal of the American Statistical Association* 58, 216–230.

L. Sachs (1997), *Angewandte Statistik*. Berlin: Springer.

## See Also

[friedmanTest](#)

## Examples

```
## Sachs (1997), pp. 671 ff.  
## 9 reviewers (blocks)  
## assigned ranks to 4 objects (groups).  
data(reviewers)  
## See Sachs (1997) p. 677  
pageTest(reviewers, alternative = "greater")
```

---

Pentosan

*Pentosan Dataset*

---

## Description

A benchmark dataset of an interlaboratory study for determining the precision of a test method on several levels of the material Pentosan.

## Format

A data frame with 189 obs. of 3 variables:

**value** numeric, test result (no unit specified)

**lab** factor, identifier of the lab (1–7)

**material** factor, identifier of the level of the material (A–I)

## Source

Tab. 8, Practice E 691, 2005, *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

---

plot.mandel	<i>Plotting mandel Objects</i>
-------------	--------------------------------

---

**Description**

Plotting method for objects inheriting from class "mandel".

**Usage**

```
## S3 method for class 'mandel'  
plot(x, alpha = 0.005, ...)
```

**Arguments**

x	an object with class "mandel".
alpha	level of significance. Defaults to 0.005.
...	further arguments, currently ignored.

**See Also**

demo(Pentosan)

**Examples**

```
##  
## Not run:  
data(Pentosan)  
md <- mandelkTest(value ~ lab, Pentosan, subset = (material == "B"))  
plot(md)  
  
## End(Not run)
```

---

plot.PMCMR	<i>Plotting PMCMR Objects</i>
------------	-------------------------------

---

**Description**

Plotting method for objects inheriting from class "PMCMR".

**Usage**

```
## S3 method for class 'PMCMR'  
plot(x, alpha = 0.05, ...)
```



**Arguments**

x                    an object of class "PMCMR".  
 alpha                the selected alpha-level. Defaults to 0.05.  
 ...                  further arguments for method boxplot.

**Value**

A box-whisker plot for each factor level. The range of the whiskers indicate the extremes (boxplot = x, ..., range=0). Letter symbols are depicted on top of each box. Different letters indicate significant differences between groups on the selected level of alpha.

**See Also**

[boxplot](#)

**Examples**

```
## data set InsectSprays
ans <- kwAllPairsNemenyiTest(count ~ spray, data = InsectSprays)
plot(ans)
plot(ans, col="red",main="My title", xlab="Spray", "Count")
```

---

powerMCTests	<i>Power Simulation for One-Factorial All-Pairs and Many-To-One Comparison Tests</i>
--------------	--

---

**Description**

Performs power simulation for one-factorial all-pairs and Many-To-One comparison tests.

**Usage**

```
powerMCTests(mu, n = 10, errfn = c("Normal", "Lognormal", "Exponential",
  "Chisquare", "TDist", "Cauchy", "Weibull"), parms = list(mean = 0, sd = 1),
  test = c("kwManyOneConoverTest", "kwManyOneDunnTest", "kwManyOneNdwTest",
  "vanWaerdenManyOneTest", "normalScoresManyOneTest", "dunnettTest",
  "tamhaneDunnettTest", "ManyOneUTest", "kwAllPairsNemenyiTest",
  "kwAllPairsDunnTest", "kwAllPairsConoverTest", "normalScoresAllPairsTest",
  "vanWaerdenAllPairsTest", "dscfAllPairsTest", "gamesHowellTest", "lsdTest",
  "scheffeTest", "tamhaneT2Test", "tukeyTest", "dunnettT3Test",
  "pairwise.t.test", "pairwise.wilcox.test", "adManyOneTest", "adAllPairsTest",
  "bwsManyOneTest", "bwsAllPairsTest"), alternative = c("two.sided",
  "greater", "less"), p.adjust.method = c("single-step", p.adjust.methods),
  alpha = 0.05, FWER = TRUE, replicates = 1000)
```

**Arguments**

mu	numeric vector of group means.
n	number of replicates per group. If n is a scalar, then a balanced design is assumed. Otherwise, n must be a vector of same length as mu.
errfn	the error function. Defaults to "Normal".
parms	a list that denotes the arguments for the error function. Defaults to list(mean=0, sd=1).
test	the multiple comparison test for which the power analysis is to be performed. Defaults to "kwManyOneConoverTest".
alternative	the alternative hypothesis. Defaults to "two.sided", ignored if the selected error function does not use this argument.
p.adjust.method	method for adjusting p values (see <a href="#">p.adjust</a> ).
alpha	the nominal level of Type I Error.
FWER	logical, indicates whether the family-wise error should be computed. Defaults to TRUE.
replicates	the number of Monte Carlo replicates or runs. Defaults to 1000.

**Details**

The linear model of a one-way ANOVA can be written as:

$$X_{ij} = \mu_i + \epsilon_{ij}$$

For each Monte Carlo run, the function simulates  $\epsilon_{ij}$  based on the given error function and the corresponding parameters. Then the specified all-pairs or many-to-one comparison test is performed. Finally, several effect sizes (Cohen's f and R-squared), error rates (per comparison error rate, false discovery rate and familywise error rate) and test powers (any-pair power, average per-pair power and all-pairs power) are calculated.

**Value**

An object with class powerPMCMR.

**Examples**

```
## Not run:
mu <- c(0, 0, 1, 2)
n <- c(5, 4, 5, 5)
set.seed(100)
powerMCTests(mu, n, errfn="Normal",
  parms=list(mean=0, sd=1),
  test="dunnettTest", replicates=1E4)

powerMCTests(mu, n, errfn="Normal",
  parms=list(mean=0, sd=1),
  test="kwManyOneDunnTest", p.adjust.method = "bonferroni",
```

```

replicates=1E4)

## End(Not run)

```

---

powerOneWayTests

*Power Simulation for One-Factorial Single Hypothesis Tests*


---

## Description

Performs power simulation for one-factorial single hypothesis tests.

## Usage

```

powerOneWayTests(mu, n = 10, errfn = c("Normal", "Lognormal", "Exponential",
  "Chisquare", "TDist", "Cauchy", "Weibull"), parms = list(mean = 0, sd = 1),
  test = c("kruskalTest", "leTest", "vanWaerdenTest", "normalScoresTest",
  "spearmanTest", "cuzickTest", "jonckheereTest", "johnsonTest", "oneway.test",
  "adKSampleTest", "bwsKSampleTest", "bwsTrendTest", "mackWolfeTest"),
  alternative = c("two.sided", "greater", "less"), var.equal = TRUE,
  dist = NULL, alpha = 0.05, FWER = TRUE, replicates = 1000, p = NULL)

```

## Arguments

mu	numeric vector of group means.
n	number of replicates per group. If n is a scalar, then a balanced design is assumed. Otherwise, n must be a vector of same length as mu.
errfn	the error function. Defaults to "Normal".
parms	a list that denotes the arguments for the error function. Defaults to list(mean=0, sd=1).
test	the test for which the power analysis is to be performed. Defaults to "kwManyOneConoverTest".
alternative	the alternative hypothesis. Defaults to "two.sided", ignored if the selected error function does not use this argument.
var.equal	a logical variable indicating whether to treat the variances in the samples as equal. "TRUE", then a simple F test for the equality of means in a one-way analysis of variance is performed. If "FALSE", an approximate method of Welch (1951) is used, which generalizes the commonly known 2-sample Welch test to the case of arbitrarily many samples. Defaults to "TRUE"; only relevant, if test = "oneway.test", otherwise ignored.
dist	the test distribution. Only relevant for <code>kruskalTest</code> . Defaults's to NULL.
alpha	the nominal level of Type I Error.
FWER	logical, indicates whether the family-wise error should be computed. Defaults to TRUE.
replicates	the number of Monte Carlo replicates or runs. Defaults to 1000.
p	the a-priori known peak as an ordinal number of the treatment group including the zero dose level, i.e. $p = \{1, \dots, k\}$ . Defaults to NULL. Only relevant, if "mackWolfeTest" is selected.

**Details**

The linear model of a one-way ANOVA can be written as:

$$X_{ij} = \mu_i + \epsilon_{ij}$$

For each Monte Carlo run, the function simulates  $\epsilon_{ij}$  based on the given error function and the corresponding parameters. Then the specified test is performed. Finally, Type I and Type II error rates are calculated.

**Value**

An object with class `powerOneWayPMCMR`.

**See Also**

[powerMCTests](#), [pwr.anova.test](#), [power.anova.test](#)

**Examples**

```
## Not run:
set.seed(12)
mu <- c(0, 0, 1, 2)
n <- c(5, 4, 5, 5)
parms <- list(mean=0, sd=1)
powerOneWayTests(mu, n, parms, test = "cuzickTest",
  alternative = "two.sided", replicates = 1E4)

## Compare power estimation for
## one-way ANOVA with balanced design
## as given by functions
## power.anova.test, pwr.anova.test
## and powerOneWayTest

groupmeans <- c(120, 130, 140, 150)
SEsq <- 500 # within-variance
n <- 10
k <- length(groupmeans)
df <- n * k - k
SSQ.E <- SEsq * df
SSQ.A <- n * var(groupmeans) * (k - 1)
sd.errfn <- sqrt(SSQ.E / (n * k - 1))
R2 <- c("R-squared" = SSQ.A / (SSQ.A + SSQ.E))
cohensf <- sqrt(R2 / (1 - R2))
names(cohensf) <- "Cohens f"

## R stats power function
power.anova.test(groups = k,
  between.var = var(groupmeans),
  within.var = SEsq,
  n = n)
```

```

## pwr power function
pwr.anova.test(k = k, n = n, f = cohensf, sig.level=0.05)

## this Monte-Carlo based estimation
set.seed(200)
powerOneWayTests(mu = groupmeans,
                 n = n,
                 parms = list(mean=0, sd=sd.errfn),
                 test = "oneway.test",
                 var.equal = TRUE,
                 replicates = 5E3)

## Compare with effect sizes
R2
cohensf

## End(Not run)

```

---

```
print.gesdTest      gesdTest Printing
```

---

### Description

print.gesdTest is the *gesdTest* method of the generic `print` function which prints its argument and returns it *invisibly* (via `invisible(x)`).

### Usage

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

### Arguments

x	an object used to select a method.
...	further arguments. Currently ignored.

---

```
print.mandel      Mandel Printing
```

---

### Description

print.mandel is the *mandel* method of the generic `print` function which prints its argument and returns it *invisibly* (via `invisible(x)`).

**Usage**

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

**Arguments**

x                    an object used to select a method.  
 ...                  further arguments. Currently ignored.

**See Also**

[mandelhTest](#), [mandelkTest](#)

print.PCMR

*PMCMR Printing*

**Description**

print.PCMR is the *PMCMR* method of the generic [print](#) function which prints its argument and returns it *invisibly* (via [invisible\(x\)](#)).

**Usage**

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

**Arguments**

x                    an object used to select a method.  
 ...                  further arguments. Currently ignored.

print.powerOneWayPMCMR

*PowerOneWayPMCMR Printing*

**Description**

print.powerOneWayPMCMR is the *powerOneWayPMCMR* method of the generic [print](#) function which prints its argument and returns it *invisibly* (via [invisible\(x\)](#)).

**Usage**

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

**Arguments**

- x an object used to select a method.
- ... further arguments. Currently ignored.

---

print.powerPMCMR	<i>PowerPMCMR Printing</i>
------------------	----------------------------

---

**Description**

print.powerPMCMR is the *powerPMCMR* method of the generic `print` function which prints its argument and returns it *invisibly* (via `invisible(x)`).

**Usage**

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

**Arguments**

- x an object used to select a method.
- ... further arguments. Currently ignored.

**See Also**

[powerMCTests](#), [powerOneWayTests](#)

---

qPCR	<i>qPCR Curve Analysis Methods</i>
------	------------------------------------

---

**Description**

The data set contains 4 classifiers (blocks), i.e. bias, linearity, precision and resolution, for 11 different qPCR analysis methods. The null hypothesis is that there is no preferred ranking of the method results per gene for the performance parameters analyzed. The rank scores were obtained by averaging results across a large set of 69 genes in a biomarker data file.

**Format**

A data frame with 4 observations on the following 11 variables.

**Cy0** a numeric vector

**LinRegPCR** a numeric vector

**Standard\_Cq** a numeric vector

**PCR\_Minor** a numeric vector

**MAK2** a numeric vector

**LRE\_E100** a numeric vector

**5PSM** a numeric vector

**DART** a numeric vector

**FPLM** a numeric vector

**LRE\_Emax** a numeric vector

**FPK\_PCR** a numeric vector

**Source**

Data were taken from Table 2 of Ruijter et al. (2013, p. 38). See also Eisinga et al. (2017, pp. 14–15).

**References**

R. Eisinga, T Heskes, B. Pelzer, M. Te Grotenhuis (2017) Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers. *BMC Bioinformatics*, 18:68.

J. M. Ruijter et al. (2013) Evaluation of qPCR curve analysis methods for reliable biomarker discovery: Bias, resolution, precision, and implications. *Methods* 59, 32–46.

---

quadeAllPairsTest	<i>All-Pairs Comparisons for Unreplicated Blocked Data (Quade's All-Pairs Test)</i>
-------------------	---

---

**Description**

Performs Quade multiple-comparison test for unreplicated blocked data.

**Usage**

```
quadeAllPairsTest(y, ...)
```

```
## Default S3 method:
```

```
quadeAllPairsTest(y, groups, blocks, dist = c("TDist",
  "Normal"), p.adjust.method = p.adjust.methods, ...)
```



**Arguments**

<code>y</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>groups</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>blocks</code>	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>dist</code>	the test distribution. Defaults to "TDist".
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>...</code>	further arguments to be passed to or from methods.

**Details**

For all-pairs comparisons of unreplicated blocked data Quade's test can be applied. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \theta_i = \theta_j$  is tested in the two-tailed test against the alternative  $A_{ij} : \theta_i \neq \theta_j, i \neq j$ .

The function has included two methods for approximate p-value estimation:

**TDist** p-values are computed from the t distribution

**Normal** p-values are computed from the standard normal distribution

If no p-value adjustment is performed (`p.adjust.method = "none"`), then a simple protected test is recommended, i.e. all-pairs comparisons should only be applied after a significant [quade.test](#). However, any method as implemented in [p.adjust.methods](#) can be selected by the user.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**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](#), [friedmanTest](#)

**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, LETTERS[1:6]))
print(y)

## Global test
quade.test(y)

## All-pairs comparisons
quadeAllPairsTest(y, dist="TDist", p.adjust.method="holm")
```

---

reviewers

*Reviewers*

---

**Description**

9 reviewers (blocks) assigned ranks to 4 objects (groups).

**Format**

The format is a 9 x 4 Matrix with Friedman type rankings:

**rows** reviewers, 1, 2, ..., 9

**columns** groups, A, B, ..., D

**Source**

Sachs (1997), p. 671 ff.

**References**

L. Sachs (1997) *Angewandte Statistik*, New York: Springer.

**Examples**

```
data(reviewers)
friedmanTest(reviewers)
pageTest(reviewers)
frdAllPairsExactTest(reviewers, p.adjust = "bonferroni")
```

---

 scheffeTest

*Scheffe's Test*


---

**Description**

Performs Scheffe's all-pairs comparisons test for normally distributed data with equal group variances.

**Usage**

```
scheffeTest(x, ...)

## Default S3 method:
scheffeTest(x, g, ...)

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

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with normally distributed residuals and equal variances Scheffe's test can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

The p-values are computed from the F-distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

J. Bortz (1993) *Statistik für Sozialwissenschaftler*. 4. Aufl., Berlin: Springer.

L. Sachs (1997) *Angewandte Statistik*, New York: Springer.

H. Scheffe (1953) A Method for Judging all Contrasts in the Analysis of Variance. *Biometrika* 40, 87–110.

**See Also**

[FDist](#), [tukeyTest](#)

**Examples**

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1, 25)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 = varN
anova(fit)
summary(scheffeTest(x, g))
```

---

shirleyWilliamsTest    *Shirley-Williams Test*

---

**Description**

Performs Shirley's nonparametric equivalent of William's test for contrasting increasing dose levels of a treatment.

**Usage**

```
shirleyWilliamsTest(x, ...)

## Default S3 method:
shirleyWilliamsTest(x, g, nperm = 1000, ...)

## S3 method for class 'formula'
shirleyWilliamsTest(formula, data, subset, na.action,
  nperm = 1000, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
nperm	number of permutations for the asymptotic permutation test. Defaults to 1000.
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 <a href="#">model.frame</a> ) 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").

**Details**

The Shirley-William test is a non-parametric step-down trend test for testing several treatment levels with a zero control. Let there be  $k$  groups including the control and let the zero dose level be indicated with  $i = 0$  and the highest dose level with  $i = m$ , then the following  $m = k - 1$  hypotheses are tested:

$$\begin{array}{ll}
 H_m : \theta_0 = \theta_1 = \dots = \theta_m, & A_m = \theta_0 \leq \theta_1 \leq \dots \leq \theta_m, \theta_0 < \theta_m \\
 H_{m-1} : \theta_0 = \theta_1 = \dots = \theta_{m-1}, & A_{m-1} = \theta_0 \leq \theta_1 \leq \dots \leq \theta_{m-1}, \theta_0 < \theta_{m-1} \\
 \vdots & \vdots \\
 H_1 : \theta_0 = \theta_1, & A_1 = \theta_0 < \theta_1
 \end{array}$$

The procedure starts from the highest dose level ( $m$ ) to the the lowest dose level (1) and stops at the first non-significant test. The consequent lowest effect dose is the treatment level of the previous test number.

The p-values are estimated through an asymptotic boot-strap method. The p-values for  $H_1$  are calculated from the t distribution with infinite degree of freedom. This function has included the modifications as recommended by Williams (1986).

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**Note**

One may increase the number of permutations to e.g. `nperm = 10000` in order to get more precise p-values. However, this will be on the expense of computational time.

**References**

Shirley, E., (1977), Nonparametric Equivalent of Williams Test for Contrasting Increasing Dose Levels of a Treatment. *Biometrics*, 33, 386–389.

Williams, D.A. (1986), Note on Shirley's nonparametric test for comparing several dose levels with a zero-dose control. *Biometrics* 42, 183–186.

**See Also**

[sample](#)

**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 <- gl(3,5)
levels(g) <- c("0", "I", "II")

## Shirley-Williams Test
shirleyWilliamsTest(x ~ g)
```

---

skillingsMackTest	<i>Skillings-Mack Test</i>
-------------------	----------------------------

---

### Description

Performs Skillings-Mack rank sum test for partially balanced incomplete block designs or partially balanced random block designs. The null hypothesis  $H_0 : \theta_i = \theta_j$  ( $i \neq j$ ) is tested against the alternative  $H_A : \theta_i \neq \theta_j$ , with at least one inequality being strict.

### Usage

```
skillingsMackTest(y, ...)

## Default S3 method:
skillingsMackTest(y, groups, blocks, ...)
```

### Arguments

y	a numeric vector of data values, or a list of numeric data vectors.
groups	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
blocks	a vector or factor object giving the block for the corresponding elements of "x". Ignored with a warning if "x" is a list.
...	further arguments to be passed to or from methods.

### Details

The function has implemented the test of Skillings and Mack (1981). The test statistic is asymptotically chi-squared distributed with  $df = k - 1$  degrees of freedom.

### Value

A list with class "htest" containing the following components:

- method** a character string indicating what type of test was performed.
- data.name** a character string giving the name(s) of the data.
- statistic** the estimated quantile of the test statistic.
- p.value** the p-value for the test.
- parameter** the parameters of the test statistic, if any.
- alternative** a character string describing the alternative hypothesis.
- estimates** the estimates, if any.
- null.value** the estimate under the null hypothesis, if any.

**Note**

The input vector/matrix 'y' must contain NA.

**References**

Skillings, J.H. and Mack, G.A. (1981) On the use of a Friedman-type statistic in balanced and unbalanced block designs. *Technometrics*. 1981; 23:171–177.

**See Also**

[friedmanTest](#), [durbinTest](#)

**Examples**

```
## Example from Hollander and Wolfe 1999,
## originally appeared in Brady 1969.
x <- cbind(c(3,1,5,2,0,0,0,0),
           c(5,3,4,NA,2,2,3,2),
           c(15,18,21,6,17,10,8,13))
colnames(x) <- c("R", "A", "B")
rownames(x) <- 1:8
skillingsMackTest(x)

## Compare with Friedman Test for CRB
## 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)
friedmanTest(y)
skillingsMackTest(y)
```

---

spearmanTest

*Testing against Ordered Alternatives (Spearman Test)*


---

**Description**

Performs a Spearman type test for testing against ordered alternatives.



**Usage**

```
spearmanTest(x, ...)

## Default S3 method:
spearmanTest(x, g, alternative = c("two.sided", "greater",
  "less"), ...)

## S3 method for class 'formula'
spearmanTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "two.sided".
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 <a href="#">model.frame</a> ) 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").

**Details**

The null hypothesis,  $H_0 : \theta_1 = \theta_2 = \dots = \theta_k$  is tested against a simple order hypothesis,  $H_A : \theta_1 \leq \theta_2 \leq \dots \leq \theta_k, \theta_1 < \theta_k$ .

The p-values are estimated from the t distribution.

**Value**

A list with class "htest" containing the following components:

**method** a character string indicating what type of test was performed.  
**data.name** a character string giving the name(s) of the data.  
**statistic** the estimated quantile of the test statistic.  
**p.value** the p-value for the test.  
**parameter** the parameters of the test statistic, if any.  
**alternative** a character string describing the alternative hypothesis.  
**estimates** the estimates, if any.  
**null.value** the estimate under the null hypothesis, if any.

## References

Kloke, J., McKean, J. W. (2015). *Nonparametric statistical methods using R*. Boca Raton, FL: Chapman & Hall/CRC.

## See Also

[kruskalTest](#) and [shirleyWilliamsTest](#) of the package **PMCMRplus**, [kruskal.test](#) of the library **stats**.

## 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 <- gl(3,5)
levels(g) <- c("A", "B", "C")

## Chacko's test
chackoTest(x, g)

## Cuzick's test
cuzickTest(x, g)

## Johnson-Mehrotra test
johnsonTest(x, g)

## Jonckheere-Terpstrata test
jonckheereTest(x, g)

## Le's test
leTest(x, g)

## Spearman type test
spearmanTest(x, g)

## Murakami's BWS trend test
bwsTrendTest(x, g)
```

---

summary.gesdTest

*Summarize an gesdTest Object*

---

## Description

Summarize an object of class *gesdTest*.

**Usage**

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

**Arguments**

object            an object of class "gesdTest".  
 ...              further arguments. Currently ignored.

---

summary.mandel	<i>Object Summary for class "mandel"</i>
----------------	--

---

**Description**

summary.mandel is a function used to produce result summaries of the results of the functions [mandelhTest](#) or [mandelkTest](#).

**Usage**

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

**Arguments**

object            an object of class "mandel" for which a summary is desired.  
 ...              further arguments. Currently ignored.

**See Also**

[mandelhTest](#), [mandelkTest](#)

---

summary.PMCMR	<i>Summarize an PMCMR Object</i>
---------------	----------------------------------

---

**Description**

Summarize an object of class *PMCMR*.

**Usage**

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

**Arguments**

object            an object of class "PMCMR".  
 ...               further arguments. Currently ignored.

**Value**

A detailed output of all pairwise hypotheses, the test statistics, the corresponding p-values and symbols that indicates the level of significance.

**See Also**

[print.PMCMR](#), [summaryGroup](#).

**Examples**

```
ans <- vanWaerdenAllPairsTest(count ~ spray, InsectSprays)
summary(ans)
```

---

summaryGroup

*Grouped Summary of an PMCMR Object*

---

**Description**

Performs a grouped summary on an PMCMR object.

**Usage**

```
summaryGroup(x, alpha = 0.05, ...)
```

**Arguments**

x                    an object of class "PMCMR".  
 alpha                the selected alpha-level. Defaults to 0.05.  
 ...                   further arguments. Currently ignored.

**Value**

Provides summary statistics for each factor level and a letter symbol, whereas different letters indicate significant differences between factor levels based on the selected level of alpha.

**See Also**

[summary.PMCMR](#)

---

tamhaneDunnnettTest      *Tamhane-Dunnnett Many-to-One Comparison Test*


---

## Description

Performs Tamhane-Dunnnett's multiple comparisons test with one control. For many-to-one comparisons in an one-factorial layout with normally distributed residuals and unequal variances Tamhane-Dunnnett's test can be used. A total of  $m = k - 1$  hypotheses can be tested. The null hypothesis  $H_i : \mu_0(x) = \mu_i(x)$  is tested in the two-tailed test against the alternative  $A_i : \mu_0(x) \neq \mu_i(x)$ ,  $1 \leq i \leq k - 1$ .

The p-values for the test are calculated from the multivariate t distribution as implemented in the function [pmvt](#).

## Usage

```
tamhaneDunnnettTest(x, ...)

## Default S3 method:
tamhaneDunnnettTest(x, g, alternative = c("two.sided",
  "greater", "less"), ...)

## S3 method for class 'formula'
tamhaneDunnnettTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"), ...)
```

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to "two.sided".
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 <a href="#">model.frame</a> ) 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> .

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

OECD (ed. 2006), *Current approaches in the statistical analysis of ecotoxicity data: A guidance to application - Annexes*. OECD Series on testing and assessment, No. 54.

**See Also**

[pmvt](#)

**Examples**

```
set.seed(245)
mn <- c(1, 2, 2^2, 2^3, 2^4)
x <- rep(mn, each=5) + rnorm(25)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g - 1)
shapiro.test(residuals(fit))
bartlett.test(x ~ g - 1)
anova(fit)
summary(tamhaneDunnettTest(x, g, alternative = "greater"))
```

---

tamhaneT2Test

*Tamhane's T2 Test*

---

**Description**

Performs Tamhane's T2 all-pairs comparison test for normally distributed data with unequal variances.

**Usage**

```
tamhaneT2Test(x, ...)

## Default S3 method:
tamhaneT2Test(x, g, ...)

## S3 method for class 'formula'
tamhaneT2Test(formula, data, subset, na.action, ...)
```

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with normally distributed residuals but unequal groups variances the T2 test of Tamhane can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x), i \neq j$ .

The p-values are computed from the t-distribution and adjusted according to Dunn-Sidak.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

A. C. Tamhane (1979), A Comparison of Procedures for Multiple Comparisons of Means with Unequal Variances. *Journal of the American Statistical Association*, 74, 471–480.

## See Also

[dunnettT3Test](#)

## Examples

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1:5, each=5)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 != varN
anova(fit)
summary(tamhaneT2Test(x, g))
```

---

trout

*Data from a Dose-Response Experiment with Trouts*

---

## Description

This data set contains results from a dose-response experiment with trouts. The experiment was conducted with five doses of 10, 25, 60, 150 and 1000 ppm, respectively, plus a zero-dose control. The response is trout weight in mg.

## Format

A data frame with 65 observations on the following 5 variables.

**CONC** a numeric vector of dose concentration in ppm

**DOSE** a factor with levels 1 2 3 4 5 6

**REPA** a factor with levels 1 2

**REPC** a factor with levels 1 2

**Y** a numeric vector of trout weight in mg

## Source

ENV/JM/MONO(2006)18/ANN, page 113.



## References

OECD (ed. 2006), *Current approaches in the statistical analysis of ecotoxicity data: A guidance to application - Annexes*. OECD Series on testing and assessment, No. 54, (ENV/JM/MONO(2006)18/ANN).

---

tukeyTest	<i>Tukey's Test</i>
-----------	---------------------

---

## Description

Performs Tukey's all-pairs comparisons test for normally distributed data with equal group variances.

## Usage

```
tukeyTest(x, ...)

## Default S3 method:
tukeyTest(x, g, ...)

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

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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> .

## Details

For all-pairs comparisons in an one-factorial layout with normally distributed residuals and equal variances Tukey's test can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

The p-values are computed from the Tukey-distribution.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

**References**

L. Sachs (1997) *Angewandte Statistik*, New York: Springer.

J. Tukey (1949) Comparing Individual Means in the Analysis of Variance, *Biometrics* 5, 99–114.

**See Also**

[Tukey, TukeyHSD](#)

**Examples**

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1, 25)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 = varN
anova(fit)
summary(tukeyTest(x, g))
```

---

uryWigginsHochbergTest

*Ury, Wiggins, Hochberg Test*

---

**Description**

Performs Ury-Wiggins and Hochberg's all-pairs comparison test for normally distributed data with unequal variances.

**Usage**

```

uryWigginsHochbergTest(x, ...)

## Default S3 method:
uryWigginsHochbergTest(x, g,
  p.adjust.method = p.adjust.methods, ...)

## S3 method for class 'formula'
uryWigginsHochbergTest(formula, data, subset, na.action,
  p.adjust.method = p.adjust.methods, ...)

```

**Arguments**

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>formula</code>	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) containing the variables in the formula formula. 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with normally distributed residuals but unequal groups variances the tests of Ury-Wiggins and Hochberg can be performed. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : \mu_i(x) = \mu_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : \mu_i(x) \neq \mu_j(x)$ ,  $i \neq j$ .

The p-values are computed from the t-distribution. The type of test depends on the selected p-value adjustment method (see also [p.adjust](#)):

**bonferroni** the Ury-Wiggins test is performed

**hochberg** the Hochberg test is performed.

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

Y. Hochberg (1976) A Modification of the T-Method of Multiple Comparisons for a One-Way Layout With Unequal Variances, *Journal of the American Statistical Association*, 71, 200–203.

H. Ury and A. D. Wiggins (1971) Large Sample and Other Multiple Comparisons Among Means, *British Journal of Mathematical and Statistical Psychology*, 24, 174–194.

## See Also

[dunnettT3Test](#)

## Examples

```
set.seed(245)
mn <- rep(c(1, 2^(1:4)), each=5)
sd <- rep(1:5, each=5)
x <- mn + rnorm(25, sd = sd)
g <- factor(rep(1:5, each=5))

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g) # var1 != varN
anova(fit)
summary(uryWigginsHochbergTest(x, g))
```

---

vanWaerdenAllPairsTest

*van-der-Waerden's All-Pairs Comparison Normal Scores Test*

---

## Description

Performs van-der-Waerden all-pairs comparison normal scores test.

**Usage**

```
vanWaerdenAllPairsTest(x, ...)

## Default S3 method:
vanWaerdenAllPairsTest(x, g,
  p.adjust.method = c("single-step", p.adjust.methods), ...)

## S3 method for class 'formula'
vanWaerdenAllPairsTest(formula, data, subset, na.action,
  p.adjust.method = c("single-step", p.adjust.methods), ...)
```

**Arguments**

<code>x</code>	a numeric vector of data values, or a list of numeric data vectors.
<code>...</code>	further arguments to be passed to or from methods.
<code>g</code>	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>formula</code>	a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) containing the variables in the formula formula. 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> .

**Details**

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals van der-Waerden's normal scores transformation can be used prior to an all-pairs comparison test. A total of  $m = k(k - 1)/2$  hypotheses can be tested. The null hypothesis  $H_{ij} : F_i(x) = F_j(x)$  is tested in the two-tailed test against the alternative  $A_{ij} : F_i(x) \neq F_j(x)$ ,  $i \neq j$ . For `p.adjust.method = "single-step"` the Tukey's studentized range distribution is used to calculate p-values (see [Tukey](#)). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by [p.adjust](#).

**Value**

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

## References

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

B. L. van der Waerden (1952) Order tests for the two-sample problem and their power, *Indagationes Mathematicae*, 14, 453–458.

## See Also

[vanWaerdenTest](#), [vanWaerdenManyOneTest](#), [normOrder](#).

---

vanWaerdenManyOneTest *van-der-Waerden's Many-One Comparisons Normal Scores Test*

---

## Description

Performs van-der-Waerden's multiple comparison normal scores test with one control.

## Usage

```
vanWaerdenManyOneTest(x, ...)

## Default S3 method:
vanWaerdenManyOneTest(x, g, alternative = c("two.sided",
  "greater", "less"), p.adjust.method = c("single-step", p.adjust.methods),
  ...)

## S3 method for class 'formula'
vanWaerdenManyOneTest(formula, data, subset, na.action,
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = c("single-step", p.adjust.methods), ...)
```

## Arguments

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
alternative	the alternative hypothesis. Defaults to two.sided.

<code>p.adjust.method</code>	method for adjusting p values (see <a href="#">p.adjust</a> ).
<code>formula</code>	a formula of the form <code>response ~ group</code> where <code>response</code> gives the data values and <code>group</code> a vector or factor of the corresponding groups.
<code>data</code>	an optional matrix or data frame (or similar: see <a href="#">model.frame</a> ) 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> .

### Details

For many-to-one comparisons in an one-factorial layout with non-normally distributed residuals van-der-Waerden's normal scores transformation can be used prior to a many-to-one comparison test. A total of  $m = k - 1$  hypotheses can be tested. The null hypothesis  $H_i : F_0(x) = F_i(x)$  is tested in the two-tailed test against the alternative  $A_i : F_0(x) \neq F_i(x)$ ,  $1 \leq i \leq k - 1$ . For `p.adjust.method = "single-step"` the multivariate t distribution is used to calculate p-values (see [pmvt](#)). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by [p.adjust](#).

### Value

A list with class "PMCMR" containing the following components:

**method** a character string indicating what type of test was performed.

**data.name** a character string giving the name(s) of the data.

**statistic** lower-triangle matrix of the estimated quantiles of the pairwise test statistics.

**p.value** lower-triangle matrix of the p-values for the pairwise tests.

**alternative** a character string describing the alternative hypothesis.

**p.adjust.method** a character string describing the method for p-value adjustment.

**model** a data frame of the input data.

**dist** a string that denotes the test distribution.

### References

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

B. L. van der Waerden (1952) Order tests for the two-sample problem and their power, *Indagationes Mathematicae*, 14, 453–458.

### See Also

[vanWaerdenTest](#), [vanWaerdenAllPairsTest](#), [pmvt](#).

**Examples**

```
## Data set PlantGrowth
## Global test
vanWaerdenTest(weight ~ group, data = PlantGrowth)

## van-der-Waerden's many-one comparison test
ans <- vanWaerdenManyOneTest(weight ~ group,
                             data = PlantGrowth,
                             p.adjust.method = "holm")

summary(ans)
```

---

vanWaerdenTest	<i>van der Waerden's Normal Scores Test</i>
----------------	---

---

**Description**

Performs van der Waerden's normal scores test.

**Usage**

```
vanWaerdenTest(x, ...)

## Default S3 method:
vanWaerdenTest(x, g, ...)

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

**Arguments**

x	a numeric vector of data values, or a list of numeric data vectors.
...	further arguments to be passed to or from methods.
g	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning 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 <a href="#">model.frame</a> ) 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").



**Details**

For one-factorial designs with non-normally distributed residuals van der Waerden's normal scores test can be performed to test the  $H_0 : F_1(x) = F_2(x) = \dots = F_k(x)$  against the  $H_A : F_i(x) \neq F_j(x)$  ( $i \neq j$ ) with at least one strict inequality.

**Note**

A tie correction is not applied in this function.

**References**

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

B. L. van der Waerden (1952) Order tests for the two-sample problem and their power, *Indagationes Mathematicae*, 14, 453–458.

**See Also**

[kruskalTest](#), [normalScoresTest](#)

**Examples**

```
vanWaerdenTest(count ~ spray, data = InsectSprays)
```

# Index

## \*Topic **datasets**

- algae, [9](#)
- Pentosan, [103](#)
- qPCR, [111](#)
- reviewers, [114](#)
- trout, [128](#)

## \*Topic **distribution**

- Cochran, [23](#)
- Dgrubbs, [28](#)
- Grubbs, [60](#)
- Mandel-h, [86](#)
- Mandel-k, [87](#)

## \*Topic **htest**

- adAllPairsTest, [4](#)
- adManyOneTest, [7](#)
- bwsAllPairsTest, [11](#)
- bwsKSampleTest, [13](#)
- bwsManyOneTest, [15](#)
- bwsTrendTest, [17](#)
- chaAllPairsNashimotoTest, [19](#)
- chackoTest, [21](#)
- cochranTest, [24](#)
- cuzickTest, [26](#)
- doubleGrubbsTest, [29](#)
- dscfAllPairsTest, [30](#)
- dunnettT3Test, [32](#)
- dunnettTest, [33](#)
- durbinAllPairsTest, [35](#)
- frdAllPairsConoverTest, [38](#)
- frdAllPairsExactTest, [40](#)
- frdAllPairsMillerTest, [42](#)
- frdAllPairsNemenyiTest, [44](#)
- frdAllPairsSiegelTest, [46](#)
- frdManyOneDemsarTest, [48](#)
- frdManyOneExactTest, [50](#)
- frdManyOneNemenyiTest, [52](#)
- friedmanTest, [54](#)
- gamesHowellTest, [56](#)
- gesdTest, [57](#)

- goreTest, [58](#)
- grubbsTest, [61](#)
- johnsonTest, [62](#)
- jonckheereTest, [64](#)
- kwAllPairsConoverTest, [68](#)
- kwAllPairsDunnTest, [70](#)
- kwAllPairsNemenyiTest, [72](#)
- kwManyOneConoverTest, [74](#)
- kwManyOneDunnTest, [76](#)
- kwManyOneNdwTest, [78](#)
- leTest, [80](#)
- lsdTest, [82](#)
- mackWolfeTest, [84](#)
- mandelhTest, [88](#)
- mandelkTest, [90](#)
- manyOneUTest, [91](#)
- MTest, [93](#)
- normalScoresAllPairsTest, [94](#)
- normalScoresManyOneTest, [96](#)
- normalScoresTest, [97](#)
- NPMTTest, [99](#)
- osrtTest, [101](#)
- pageTest, [102](#)
- quadeAllPairsTest, [112](#)
- scheffeTest, [115](#)
- shirleyWilliamsTest, [116](#)
- skillingsMackTest, [119](#)
- spearmanTest, [120](#)
- tamhaneDunnettTest, [125](#)
- tamhaneT2Test, [126](#)
- tukeyTest, [129](#)
- uryWigginsHochbergTest, [130](#)
- vanWaerdenAllPairsTest, [132](#)
- vanWaerdenManyOneTest, [134](#)
- vanWaerdenTest, [136](#)

## \*Topic **methods**

- summary.mandel, [123](#)

## \*Topic **misc**

- powerMCTests, [105](#)

- powerOneWayTests, 107
- \*Topic **models**
  - summary.gesdTest, 122
  - summary.PMCMR, 123
  - summaryGroup, 124
- \*Topic **nonparametric**
  - adAllPairsTest, 4
  - adManyOneTest, 7
  - bwsAllPairsTest, 11
  - bwsKSampleTest, 13
  - bwsManyOneTest, 15
  - bwsTrendTest, 17
  - chaAllPairsNashimotoTest, 19
  - chackoTest, 21
  - cuzickTest, 26
  - dscfAllPairsTest, 30
  - durbinAllPairsTest, 35
  - frdAllPairsConoverTest, 38
  - frdAllPairsExactTest, 40
  - frdAllPairsMillerTest, 42
  - frdAllPairsNemenyiTest, 44
  - frdAllPairsSiegelTest, 46
  - frdManyOneDemsarTest, 48
  - frdManyOneExactTest, 50
  - frdManyOneNemenyiTest, 52
  - friedmanTest, 54
  - goreTest, 58
  - johnsonTest, 62
  - jonckheereTest, 64
  - kwAllPairsConoverTest, 68
  - kwAllPairsDunnTest, 70
  - kwAllPairsNemenyiTest, 72
  - kwManyOneConoverTest, 74
  - kwManyOneDunnTest, 76
  - kwManyOneNdwTest, 78
  - leTest, 80
  - mackWolfeTest, 84
  - normalScoresAllPairsTest, 94
  - normalScoresManyOneTest, 96
  - normalScoresTest, 97
  - NPMTTest, 99
  - pageTest, 102
  - quadeAllPairsTest, 112
  - shirleyWilliamsTest, 116
  - skillingsMackTest, 119
  - spearmanTest, 120
  - vanWaerdenAllPairsTest, 132
  - vanWaerdenManyOneTest, 134
  - vanWaerdenTest, 136
- \*Topic **nonparanetric**
  - manyOneUTest, 91
- \*Topic **plot**
  - barPlot, 10
  - plot.PMCMR, 104
- \*Topic **print**
  - print.gesdTest, 109
  - print.mandel, 109
  - print.PMCMR, 110
  - print.powerOneWayPMCMR, 110
  - print.powerPMCMR, 111
- \*Topic **univariate**
  - gesdTest, 57
- ad.pval, 5–7, 9
- adAllPairsTest, 4, 7, 9
- adKSampleTest, 5, 6, 9
- adManyOneTest, 5, 7, 7
- algae, 9
- barPlot, 10
- bartlett.test, 26
- boxplot, 105
- bws\_cdf, 16
- bws\_stat, 16
- bws\_test, 12
- bwsAllPairsTest, 11, 14, 18
- bwsKSampleTest, 13
- bwsManyOneTest, 15
- bwsTrendTest, 17
- chaAllPairsNashimotoTest, 19
- chackoTest, 19, 21, 21
- Chisquare, 68, 73
- Cochran, 23
- cochranTest, 24
- cuzickTest, 26
- default (kwAllPairsDunnTest), 70
- Dgrubbs, 28
- doubleGrubbsTest, 29
- dscfAllPairsTest, 30
- dunnettT3Test, 32, 128, 132
- dunnettTest, 33
- durbinAllPairsTest, 35
- durbinTest, 36, 37, 59, 120
- FDist, 24, 68, 116

- fligner.test, 26
- frdAllPairsConoverTest, 38, 42, 43, 45, 47
- frdAllPairsExactTest, 40, 40, 43, 45, 47
- frdAllPairsMillerTest, 40, 42, 42, 45, 47
- frdAllPairsNemenyiTest, 40, 42, 43, 44, 47
- frdAllPairsSiegelTest, 40, 42, 43, 45, 46
- frdManyOneDemSarTest, 48, 51, 53
- frdManyOneExactTest, 49, 50, 53
- frdManyOneNemenyiTest, 49, 51, 52
- friedman.test, 40, 42, 43, 45, 47, 49, 51, 53, 55
- friedmanTest, 40, 42, 43, 45, 47, 49, 51, 53, 54, 59, 103, 114, 120
  
- gamesHowellTest, 56
- gesdTest, 57
- goreTest, 58
- Grubbs, 29, 60
- grubbsTest, 61
  
- invisible, 109–111
  
- johnsonTest, 62
- jonckheereTest, 64
  
- kruskal.test, 18, 23, 27, 63, 65, 68, 82, 122
- kruskalTest, 18, 23, 27, 63, 65, 66, 70, 72, 73, 75, 78, 80, 82, 99, 107, 122, 137
- kwAllPairsConoverTest, 68, 72, 73
- kwAllPairsDunnTest, 70, 70, 73
- kwAllPairsNemenyiTest, 70, 72, 72, 99, 100
- kwManyOneConoverTest, 74, 78, 80
- kwManyOneDunnTest, 75, 76, 80
- kwManyOneNdwTest, 75, 78, 78
  
- leTest, 80
- lsdTest, 82
  
- mackWolfeTest, 84
- Mandel-h, 86
- Mandel-k, 87
- mandelhTest, 87, 88, 110, 123
- mandelkTest, 88, 90, 110, 123
- manyOneUTest, 91
- model.frame, 5, 6, 8, 11, 13, 15, 18, 20, 22, 25, 27, 31, 32, 34, 56, 62, 64, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 89, 90, 92, 93, 95, 96, 98, 100, 101, 115, 117, 121, 125, 127, 129, 131, 133, 135, 136
  
- MTest, 93
- murakami\_cdf, 16
- murakami\_stat, 16
  
- Normal, 21, 72, 92
- normalScoresAllPairsTest, 94, 97
- normalScoresManyOneTest, 95, 96
- normalScoresTest, 95, 97, 97, 137
- normOrder, 95, 97, 134
- NPMTTest, 99
  
- osrtTest, 93, 101
  
- p.adjust, 4, 5, 8, 11, 12, 15, 16, 36, 39, 41, 46–50, 69–73, 75, 77, 79, 91, 92, 94–97, 106, 113, 131, 133, 135
- p.adjust.methods, 20, 36, 113
- pageTest, 102
- pairwise.t.test, 84
- pairwise.wilcox.test, 32
- pcochran, 25
- pcochran (Cochran), 23
- pdgrubbs, 29
- pdgrubbs (Dgrubbs), 28
- Pentosan, 103
- pgrubbs, 61
- pgrubbs (Grubbs), 60
- pKruskalWallis, 67, 68
- plot.mandel, 104
- plot.PMCMR, 104
- pmandelh, 88, 89
- pmandelh (Mandel-h), 86
- pmandelk, 91
- pmandelk (Mandel-k), 87
- pmvnorm, 53, 78, 92
- pmvt, 32–35, 75, 80, 97, 125, 126, 135
- pnorm, 85, 86
- pNormScore, 98, 99
- power.anova.test, 108
- powerMCTests, 105, 108, 111
- powerOneWayTests, 107, 111
- print, 109–111
- print (print.mandel), 109
- print.gesdTest, 109
- print.mandel, 109
- print.PMCMR, 110, 124
- print.powerOneWayPMCMR, 110
- print.powerPMCMR, 111
- ptukey, 57

pwr.anova.test, 108

qcochran (Cochran), 23

qgrubbs (Grubbs), 60

qmandelh, 88, 89

qmandelh (Mandel-h), 86

qmandelk, 91

qmandelk (Mandel-k), 87

qPCR, 111

quade.test, 113, 114

quadeAllPairsTest, 112

reviewers, 114

sample, 14, 18, 86, 118

scheffeTest, 115

shirleyWilliamsTest, 18, 23, 27, 63, 65, 82,  
116, 122

skillingsMackTest, 59, 119

spearmanTest, 120

summary.gesdTest, 122

summary.mandel, 123

summary.PCMR, 123, 124

summaryGroup, 124, 124

tamhaneDunnnettTest, 125

tamhaneT2Test, 126

TDist, 60, 70, 75, 78, 80, 84

trout, 128

Tukey, 21, 32, 70, 73, 95, 130, 133

TukeyHSD, 130

tukeyTest, 116, 129

uryWigginsHochbergTest, 130

vanWaerdenAllPairsTest, 132, 135

vanWaerdenManyOneTest, 134, 134

vanWaerdenTest, 99, 134, 135, 136

vanWeardenTest.default  
(vanWaerdenTest), 136

wilcox.test, 92