Package ‘PMCMRplus’

September 27, 2020

Type Package

Title Calculate Pairwise Multiple Comparisons of Mean Rank Sums

Extended

Version 1.5.1

Date 2020-09-26

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), BWSfast (>= 0.2.1), MASS

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

**SystemRequirements**  gmp (>= 4.2.3), mpfr (>= 3.0.0) or file README.md

**SystemRequirementsNote**  see >> README.md

**SysDataCompression**  gzip

**VignetteBuilder**  knitr, rmarkdown

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

**NeedsCompilation**  yes

**Encoding**  UTF-8

**LazyData**  true

**RoxygenNote**  7.1.1

**License**  GPL (>= 3)

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

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

**Repository**  CRAN

**Date/Publication**  2020-09-27 17:10:26 UTC

---

**R topics documented:**

- adAllPairsTest .......................................................... 4
- adKSampleTest ........................................................... 6
- adManyOneTest ........................................................... 8
- algae ........................................................................ 10
- barPlot ....................................................................... 11
- bwsAllPairsTest ........................................................... 12
- bwsKSampleTest ........................................................... 14
- bwsManyOneTest ......................................................... 16
- bwsTrendTest ............................................................. 18
- chaAllPairsNashimotoTest .............................................. 21
- chackoTest ................................................................. 23
- Cochran ................................................................. 25
- cochrantTest ............................................................... 26
- cuzickTest ................................................................. 28
- Dgrubbs ................................................................. 30
- doubleGrubbsTest ......................................................... 31
- dscfAllPairsTest ........................................................... 32
- duncanTest ............................................................... 34
- dunnettT3Test .............................................................. 35
- dunnettTest ............................................................... 37
- durbinAllPairsTest ......................................................... 39
- durbinTest ............................................................... 41
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>frdAllPairsConoverTest</td>
<td>42</td>
</tr>
<tr>
<td>frdAllPairsExactTest</td>
<td>44</td>
</tr>
<tr>
<td>frdAllPairsMillerTest</td>
<td>47</td>
</tr>
<tr>
<td>frdAllPairsNemenyiTest</td>
<td>49</td>
</tr>
<tr>
<td>frdAllPairsSiegelTest</td>
<td>52</td>
</tr>
<tr>
<td>frdManyOneDemsarTest</td>
<td>54</td>
</tr>
<tr>
<td>frdManyOneExactTest</td>
<td>56</td>
</tr>
<tr>
<td>frdManyOneNemenyiTest</td>
<td>58</td>
</tr>
<tr>
<td>friedmanTest</td>
<td>60</td>
</tr>
<tr>
<td>gamesHowellTest</td>
<td>63</td>
</tr>
<tr>
<td>gsdTest</td>
<td>64</td>
</tr>
<tr>
<td>goreTest</td>
<td>65</td>
</tr>
<tr>
<td>Grubbs</td>
<td>67</td>
</tr>
<tr>
<td>grubbsTest</td>
<td>68</td>
</tr>
<tr>
<td>GSTTest</td>
<td>69</td>
</tr>
<tr>
<td>hartleyTest</td>
<td>71</td>
</tr>
<tr>
<td>johnsonTest</td>
<td>73</td>
</tr>
<tr>
<td>jonckheereTest</td>
<td>75</td>
</tr>
<tr>
<td>kruskalTest</td>
<td>77</td>
</tr>
<tr>
<td>kwAllPairsConoverTest</td>
<td>80</td>
</tr>
<tr>
<td>kwAllPairsDunnTest</td>
<td>82</td>
</tr>
<tr>
<td>kwAllPairsNemenyiTest</td>
<td>84</td>
</tr>
<tr>
<td>kwManyOneConoverTest</td>
<td>86</td>
</tr>
<tr>
<td>kwManyOneDunnTest</td>
<td>88</td>
</tr>
<tr>
<td>kwManyOneNdwTest</td>
<td>91</td>
</tr>
<tr>
<td>leTest</td>
<td>93</td>
</tr>
<tr>
<td>lsdTest</td>
<td>95</td>
</tr>
<tr>
<td>mackWolfeTest</td>
<td>97</td>
</tr>
<tr>
<td>Mandel-h</td>
<td>99</td>
</tr>
<tr>
<td>Mandel-k</td>
<td>100</td>
</tr>
<tr>
<td>mandelhTest</td>
<td>101</td>
</tr>
<tr>
<td>mandelkTest</td>
<td>102</td>
</tr>
<tr>
<td>manyOneUTest</td>
<td>104</td>
</tr>
<tr>
<td>MTest</td>
<td>105</td>
</tr>
<tr>
<td>normalScoresAllPairsTest</td>
<td>107</td>
</tr>
<tr>
<td>normalScoresManyOneTest</td>
<td>108</td>
</tr>
<tr>
<td>normalScoresTest</td>
<td>110</td>
</tr>
<tr>
<td>NPMTest</td>
<td>112</td>
</tr>
<tr>
<td>osrTest</td>
<td>113</td>
</tr>
<tr>
<td>pageTest</td>
<td>115</td>
</tr>
<tr>
<td>Pentosan</td>
<td>116</td>
</tr>
<tr>
<td>plot.mandel</td>
<td>117</td>
</tr>
<tr>
<td>plot.PMCMR</td>
<td>117</td>
</tr>
<tr>
<td>powerMCTests</td>
<td>118</td>
</tr>
<tr>
<td>powerOneWayTests</td>
<td>120</td>
</tr>
<tr>
<td>print.gesdTest</td>
<td>122</td>
</tr>
<tr>
<td>print.mandel</td>
<td>123</td>
</tr>
<tr>
<td>print.PMCMR</td>
<td>123</td>
</tr>
</tbody>
</table>
Anderson-Darling All-Pairs Comparison Test

Performs Anderson-Darling all-pairs comparison test.

Usage

```r
adAllPairsTest(x, ...)
```

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

## S3 method for class 'formula'
```r
dummy
```
adAllPairsTest

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 p.adjust).
formula               a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data                an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset               an optional vector specifying a subset of observations to be used.
na.action             a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

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

See Also
adKSampleTest, adManyOneTest, ad.pval.

Examples
adKSampleTest(count ~ spray, InsectSprays)

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

---

adKSampleTestAnderson-Darling k-Sample Test

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 `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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

See Also
`adAllPairsTest`, `adManyOneTest`, `ad.pval`.

Examples
```r
## 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
g <- factor(x = c(rep(1, length(x)),
               rep(2, length(y))),
```
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(

```r
rep(3, length(z))),
labels = c("ns", "oad", "a")
dat <- data.frame(
  g = g,
  x = c(x, y, z))

## AD-Test
adKSampleTest(x ~ g, data = dat)

## BWS-Test
bwsKSampleTest(x ~ g, data = dat)

## Kruskal-Test
# Using incomplete beta approximation
kruskalTest(x ~ g, dat, dist="KruskalWallis")
# Using chisquare distribution
kruskalTest(x ~ g, dat, dist="Chisquare")

## Not run:
## Check with kruskal.test from R stats
kruskal.test(x ~ g, dat)

## End(Not run)
## Using Conover's F
kruskalTest(x ~ g, dat, dist="FDist")

## Not run:
## Check with aov on ranks
anova(aov(rank(x) ~ g, dat))
# Check with oneway.test
oneway.test(rank(x) ~ g, dat, var.equal = TRUE)

## End(Not run)
```
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 p.adjust).
formula  a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data  an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset  an optional vector specifying a subset of observations to be used.
na.action  a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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 ≠ F_i, (1 ≤ i ≤ 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

Statistical Association 82, 918–924.

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


References


See Also
demo(algae)

barPlot

Plotting PMCMR Objects

Description

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

Usage

```r
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 $\pm 2(1 - \alpha/2) \times \hat{SE}_i$, 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

```r
## data set chickwts
ans <- tukeyTest(weight ~ feed, data = chickwts)
barPlot(ans)
```
bwsAllPairsTest | BWS All-Pairs Comparison Test

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 p.adjust).
formula | a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data | an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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 `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


See Also

`bws_test`.

Examples

```r
out <- bwsAllPairsTest(count ~ spray, InsectSprays, p.adjust="holm")
summary(out)
summaryGroup(out)
```
Murakami’s k-Sample BWS Test

Description
Performs Murakami’s k-Sample BWS Test.

Usage
bwsKSampleTest(x, ...)

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

## S3 method for class 'formula'
bwsKSampleTest(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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
- **subset**: an optional vector specifying a subset of observations to be used.
- **na.action**: a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details
The null hypothesis, \( H_0 : F_1 = F_2 = \ldots = F_k \) 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 "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

See Also
sample, bwsAllPairsTest, bwsManyOneTest.

Examples
```r
## 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
## g <- factor(x = c(rep(1, length(x)),
##                rep(2, length(y)),
##                rep(3, length(z))),
##              labels = c("ns", "oad", "a"))
## dat <- data.frame(
##                   g = g,
##                   x = c(x, y, z))

## AD-Test
adKSampleTest(x ~ g, data = dat)

## BWS-Test
bwsKSampleTest(x ~ g, data = dat)

## Kruskal-Test
## Using incomplete beta approximation
kruskalTest(x ~ g, dat, dist="KruskalWallis")
## Using chisquare distribution
```
kruskalTest(x ~ g, dat, dist="Chisquare")

## Not run:
## Check with kruskal.test from R stats
kruskal.test(x ~ g, dat)

## End(Not run)
## Using Conover’s F
kruskalTest(x ~ g, dat, dist="FDist")

## Not run:
## Check with aov on ranks
anova(aov(rank(x) ~ g, dat))
## Check with oneway.test
oneway.test(rank(x) ~ g, dat, var.equal = TRUE)

## End(Not run)

bwsManyOneTest  
BWS Many-To-One Comparison Test

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

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.
method  a character string specifying the test statistic to use. Defaults to BWS.
p.adjust.method  method for adjusting p values (see p.adjust).
formula  a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data  an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset  an optional vector specifying a subset of observations to be used.
na.action  a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

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 bws_stat and 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 murakami_stat and 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 murakami_stat and 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**


---

**See Also**

`murakami_stat`, `murakami_cdf`, `bws_stat`, `bws_cdf`.

---

**Examples**

```r
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.
bwsTrendTest

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 model.frame) containing  
the variables in the formula formula. By default the variables are taken from  
environment(formula).

subset  
an optional vector specifying a subset of observations to be used.

na.action  
a function which indicates what should happen when the data contain NAs. De-  
faults to getOption("na.action").

Details

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


See Also

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

Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- gl(3,5)
levels(g) <- c("A", "B", "C")

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

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

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

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

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

## Spearman type test
spearmanTest(x, g)

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

```r
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

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

data an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula formula. By default the variables are taken from `environment(formula)`.

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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


See Also

- `Tukey`, `Normal`, `chackoTest`
### Example from Sachs (1997, p. 402)

```r
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**

```r
chackoTest(x, ...)
```

**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 `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- `subset` an optional vector specifying a subset of observations to be used.
- `na.action` a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

**Details**

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

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**:


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


The Algorithm AS 206 is available from StatLib [http://lib.stat.cmu.edu/apstat/](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


See Also

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

Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- 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-Terpstra test

jonckheereTest(x, g)

## Le's test

leTest(x, g)

## Spearman type test

spearmanTest(x, g)

## Murakami's BWS trend test

bwsTrendTest(x, g)

---

### Description

Distribution function and quantile function for Cochran’s distribution.

### Usage

```r
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


See Also

FDist

Examples

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

---

### cochranTest

#### Cochran Test

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

#### Usage

```r
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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details
For normally distributed data the null hypothesis, $H_0 : \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2$ 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

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 response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data: an optional matrix or data frame (or similar: see `model.frame`) containing
the variables in the formula `formula`. By default the variables are taken from
`environment(formula)`.

subset: an optional vector specifying a subset of observations to be used.

na.action: a function which indicates what should happen when the data contain NAs. De-
defaults to `getOption("na.action")`.

Details

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

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


See Also

`kruskalTest` and `shirleyWilliamsTest` of the package `PMCMRplus`, `kruskal.test` of the li-
brary `stats`.

Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- 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-Terpstra 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

```r
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 <= x]` otherwise, `P[X > x]`.
- `log.p`: logical; if TRUE, probabilities p are given as log(p).

### Value

`pdgrubbs` gives the distribution function

### References


doubleGrubbsTest

See Also

Grubbs

Examples

pdgrubbs(0.62, 7, 1E4)

doubleGrubbsTest  Grubbs Double Outlier Test

Description

Performs Grubbs double outlier test.

Usage

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

Arguments

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

Details

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

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

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

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

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

The p-value is computed with the function pdgrubbs.
dscfAllPairsTest

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


Examples

```r
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 \(\text{response} \sim \text{group}\) where \text{response}\ gives the data values and \text{group}\ a vector or factor of the corresponding groups.
- **data**: an optional matrix or data frame (or similar: see \texttt{model.frame}) containing the variables in the formula \texttt{formula}. By default the variables are taken from \texttt{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 \texttt{getOption("na.action")}.

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


See Also

Tukey, pairwise.wilcox.test

duncanTest

Duncan’s Multiple Range Test

Description

Performs Duncan’s all-pairs comparisons test for normally distributed data with equal group variances.

Usage

duncanTest(x, ...)

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

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

## S3 method for class 'aov'
duncanTest(x, ...)

Arguments

x a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.

... 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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

Details

For all-pairs comparisons in an one-factorial layout with normally distributed residuals and equal variances Duncan’s multiple range 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**


**See Also**

- Tukey, TukeyHSD tukeyTest

**Examples**

```r
fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts)
anova(fit)
```

```r
## also works with fitted objects of class aov
res <- duncanTest(fit)
summary(res)
summaryGroup(res)
```

---

**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, ...)

### S3 method for class 'aov'
dunnettT3Test(x, ...)

Arguments

x  
a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.

...  
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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  
an optional vector specifying a subset of observations to be used.

na.action  
a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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 \( m = \frac{k(k-1)}{2} \) 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

See Also
pmvt

Examples
fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts)
anova(fit)

## also works with fitted objects of class aov
res <- dunnettT3Test(fit)
summary(res)
summaryGroup(res)

---

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,
dunnettTest

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

## S3 method for class 'aov'
dunnettTest(x, alternative = c("two.sided", "greater", "less"), ...)

Arguments

x a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.
...

Further arguments to be passed to or from methods.

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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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


See Also

*pmvt*

Examples

```r
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)
## works with fitted object of class aov
summary(dunnettTest(fit, 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**

```r
durbinAllPairsTest(y, ...)
```

---

**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 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 (\( \text{p.adjust.method} = \text{"none"} \)), than a simple protected test is recommended, i.e. the all-pairs comparisons should only be applied after a significant \texttt{durbinTest}. However, any method as implemented in \texttt{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


See Also

- \texttt{durbinTest}

Examples

```r
## 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, 2, NA, 2, 1, NA, NA, NA,
             NA, 2, 1, NA, NA, NA, 3, NA, 2, 2),
nrow=7, ncol=7, byrow=FALSE, dimnames=list(1:7, LETTERS[1:7]))
durbinAllPairsTest(y)
```
**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**

- `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 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 "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 function does not test, whether it is a true BIBD. This function does not test for ties.

References


Examples

```r
## Example for an incomplete block design:
## Data from Conover (1999, p. 391).
y <- matrix(c(
  2, NA, NA, NA, 3, NA, 3, NA, NA, 3, NA, NA,
  1, 2, NA, NA, NA, 1, 1, NA, NA, 1, 1,
  NA, NA, NA, NA, 2, NA, 2, 1, NA, NA, NA, NA,
  3, NA, 2, 1, NA, NA, NA, 3, NA, 2, 2
), ncol=7, nrow=7, byrow=FALSE,
dimnames=list(1:7, LETTERS[1:7]))
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

```r
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


See Also

- `friedmanTest`
- `friedman.test`
- `frdAllPairsExactTest`
- `frdAllPairsMillerTest`
- `frdAllPairsNemenyiTest`
- `frdAllPairsSiegelTest`
Examples

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

## Siegel-Castellan
frdAllPairsSiegelTest(y=y, p.adjust = "bonferroni")

## Irrelevant of group order?
x <- as.vector(y)
g <- rep(colnames(y), each = length(x)/length(colnames(y)))
b <- rep(rownames(y), times = length(x)/length(rownames(y)))
xD <- data.frame(x, g, b) # grouped by colnames
frdAllPairsNemenyiTest(xDF$x, groups = xDF$g, blocks = xDF$b)
o <- order(xDF$b) # order per block increasingly
frdAllPairsNemenyiTest(xDF$x[o], groups = xDF$g[o], blocks = xDF$b[o])
o <- order(xDF$x) # order per value increasingly
frdAllPairsNemenyiTest(xDF$x[o], groups = xDF$g[o], blocks = xDF$b[o])

## formula method (only works for Nemenyi)
frdAllPairsNemenyiTest(x ~ g | b, data = xDF)
```

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

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, 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.
frdAllPairsExactTest

`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:


**References**


**See Also**

`friedmanTest`, `friedman.test`, `frdAllPairsConoverTest`, `frdAllPairsMillerTest`, `frdAllPairsNemenyiTest`, `frdAllPairsSiegelTest`

**Examples**

```r
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics
## (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
##
## y <- matrix(c(3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
##               23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
##               26.67, 4.44, 7.94, 4.04, 4.23, 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)
```

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

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

```r
## Nemenyi's test
frdAllPairsNemenyiTest(y=y)
```
## Miller et al.
frdAllPairsMillerTest(y=y)

## Siegel-Castellan
frdAllPairsSiegelTest(y=y, p.adjust = "bonferroni")

## Irrelevant of group order?
x <- as.vector(y)
g <- rep(colnames(y), each = length(x)/length(colnames(y)))
b <- rep(rownames(y), times = length(x)/length(rownames(y)))
xDf <- data.frame(x, g, b) # grouped by colnames

frdAllPairsNemenyiTest(xDf$x, groups = xDf$g, blocks = xDf$b)
o <- order(xDf$b) # order per block increasingly
frdAllPairsNemenyiTest(xDf$x[o], groups = xDf$g[o], blocks = xDf$b[o])
o <- order(xDf$x) # order per value increasingly
frdAllPairsNemenyiTest(xDf$x[o], groups = xDf$g[o], blocks = xDf$b[o])

## formula method (only works for Nemenyi)
frdAllPairsNemenyiTest(x ~ g | b, data = xDf)

---

frdAllPairsMillerTest  Miller’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
- **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, 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


See Also

- `friedmanTest`
- `friedman.test`
- `frdAllPairsExactTest`
- `frdAllPairsConoverTest`
- `frdAllPairsNemenyiTest`
- `frdAllPairsSiegelTest`

Examples

```r
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics
## (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
##
## y <- matrix(c(
  3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
  23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
  26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
  32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
  26.65), nrow=6, ncol=6),
```
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, ...)
## S3 method for class 'formula'
frdAllPairsNemenyiTest(formula, data, subset, na.action, ...)

### 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.
- `formula`: a formula of the form a ~ b | c where a, b and c give the data values and the corresponding groups and blocks, respectively.
- `data`: an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- `subset`: an optional vector specifying a subset of observations to be used.
- `na.action`: a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.
- `...`: 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 \( m = \frac{k(k - 1)}{2} \) 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.
frdAllPairsNemenyiTest

References


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)

## Siegel-Castellan
frdAllPairsSiegelTest(y=y, p.adjust = "bonferroni")

## Irrelevant of group order?
x <- as.vector(y)
g <- rep(colnames(y), each = length(x)/length(colnames(y)))
b <- rep(rownames(y), times = length(x)/length(rownames(y)))
xDf <- data.frame(x, g, b) # grouped by colnames
frdAllPairsNemenyiTest(xDF$x, groups = xDF$g, blocks = xDF$b)
o <- order(xDF$b) # order per block increasingly
frdAllPairsNemenyiTest(xDF$x[o], groups = xDF$g[o], blocks = xDF$b[o])
o <- order(xDF$x) # order per value increasingly
frdAllPairsNemenyiTest(xDF$x[o], groups = xDF$g[o], blocks = xDF$b[o])

## formula method (only works for Nemenyi)
frdAllPairsNemenyiTest(x ~ g | b, data = xDF)

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 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, Siegel and Castellan’s test can be performed on Friedman-type ranked data.

A total of \( m = \frac{k(k - 1)}{2} \) 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 \texttt{p.adjust} can be used for \( p \)-value adjustment.

Value

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

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

References


See Also

\texttt{friedmanTest}, \texttt{friedman.test}, \texttt{frdAllPairsExactTest}, \texttt{frdAllPairsConoverTest}, \texttt{frdAllPairsNemenyiTest}, \texttt{frdAllPairsMillerTest}

Examples

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

## Siegel-Castellan

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

## Irrelevant of group order?

x <- as.vector(y)
g <- rep(colnames(y), each = length(x)/length(colnames(y)))
b <- rep(rownames(y), times = length(x)/length(rownames(y)))
xDf <- data.frame(x, g, b) # grouped by colnames

frdAllPairsNemenyiTest(xDF$x, groups = xDF$g, blocks = xDF$b)
o <- order(xDF$b) # order per block increasingly
frdAllPairsNemenyiTest(xDF$x[o], groups = xDF$g[o], blocks = xDF$b[o])
o <- order(xDF$x) # order per value increasingly
frdAllPairsNemenyiTest(xDF$x[o], groups = xDF$g[o], blocks = xDF$b[o])

## formula method (only works for Nemenyi)

frdAllPairsNemenyiTest(x ~ g | b, data = xDF)

---

### frdManyOneDemsarTest

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

**Description**

Performs Demscar’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"),
)
```r
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 `p.adjust`).
- `...` — 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**

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

- **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 many-to-one comparisons (pairwise comparisons with one control) in a two factorial unrepli-
cated 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 \( A_i : \theta_0 \neq \theta_i, \quad (1 \leq i \leq m) \).

The exact p-values are computed using the code of "pexactfriesd.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

Eisinga, R., Heskes, T., Pelzer, B., Te Grotenhuis, M. (2017) Exact p-values for Pairwise Com-
parison 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.

```r
y <- matrix(c(
3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
26.65), nrow=6, ncol=6, dimnames=list(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)`

---

### Description

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

### Usage

```r
frdManyOneNemenyiTest(y, ...)
```

## Default S3 method:
`frdManyOneNemenyiTest(`
```r
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.

Details

For many-to-one comparisons (pairwise comparisons with one control) in a two factorial unrepli-
cated complete block design with non-normally distributed residuals, Nemenyi’s test can be per-
formed 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.

$H_0 : \theta_0 = \theta_i$ 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. As `pmvnorm` applies a numerical
method, the estimated $p$-values are set depended.

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

York: Wiley. 2014.


See Also

friedmanTest, friedman.test, frdManyOneExactTest, frdManyOneDemsarTest, pmvnorm, set.seed

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       Friedman Rank Sum Test

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 `Chisquare`.
- `...`: 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, than 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


See Also

- `friedman.test`
Examples

## 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,
       5.45, 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,
       5.95, 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, ...)
## S3 method for class 'aov'
gamesHowellTest(x, ...)

Arguments

x  a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.
...
  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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset  an optional vector specifying a subset of observations to be used.
na.action  a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

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.
gesdTest

Generalized Extreme Studentized Deviate Many-Outlier Test

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.
goreTest

References


Examples

```r
## 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


See Also

- `friedmanTest`
- `skillingsMackTest`
- `durbinTest`

Examples

```r
## 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=" ",)
close(con)
colnames(x) <- c("Yield", "Variety", "SoilType")
goreTest(y = x$Yield, groups = x$Variety, blocks = x$SoilType)
```
Grubbs distribution

Description

Distribution function and quantile function for Grubbs distribution.

Usage

\texttt{qgrubbs(p, n)}

\texttt{pgrubbs(q, n, lower.tail = TRUE)}

Arguments

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

Value

\texttt{pgrubbs} gives the distribution function and \texttt{qgrubbs} gives the quantile function.

References


See Also

\texttt{TDist}

Examples

\texttt{qgrubbs(0.05, 7)}
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 $x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$. Then the following model for a single maximum outlier can be proposed:

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

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

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

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


Examples

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

---

**GSTTest**

*Generalized Siegel-Tukey Test of Homogeneity of Scales*

Description

Performs a Siegel-Tukey k-sample rank dispersion test.

Usage

```r
GSTTest(x, ...)
```

## Default S3 method:

```r
GSTTest(x, g, dist = c("Chisquare", "KruskalWallis"), ...)
```

## S3 method for class 'formula'

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

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 \texttt{model.frame}) containing the variables in the formula \texttt{formula}. By default the variables are taken from \texttt{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 \texttt{getOption("na.action")}.

Details

Meyer-Bahlburg (1970) has proposed a generalized Siegel-Tukey rank dispersion test for the $k$-sample case. Likewise to the \texttt{fligner.test}, this test is a nonparametric test for testing the homogeneity of scales in several groups. Let $\theta_i$ and $\lambda_i$ denote location and scale parameter of the $i$th group, then for the two-tailed case, the null hypothesis $H$: $\lambda_i/\lambda_j = 1 | \theta_i = \theta_j$, $i \neq j$ is tested against the alternative, $A$: $\lambda_i/\lambda_j \neq 1$ with at least one inequality being strict.

The data are combinedly ranked according to Siegel-Tukey. The ranking is done by alternate extremes (rank 1 is lowest, 2 and 3 are the two highest, 4 and 5 are the two next lowest, etc.).

Meyer-Bahlburg (1970) showed, that the Kruskal-Wallis H-test can be employed on the Siegel-Tukey ranks. The H-statistic is asymptotically chi-squared distributed with $v = k - 1$ degree of freedom, the default test distribution is consequently \texttt{dist = "Chisquare"}. If \texttt{dist = "KruskalWallis"} is selected, an incomplete beta approximation is used for the calculation of p-values as implemented in the function \texttt{pKruskalWallis} of the package \texttt{SuppDists}.

Value

A list with class "\texttt{htest}" containing the following components:

- \texttt{method}  a character string indicating what type of test was performed.
- \texttt{data.name}  a character string giving the name(s) of the data.
- \texttt{statistic}  the estimated quantile of the test statistic.
- \texttt{p.value}  the p-value for the test.
- \texttt{parameter}  the parameters of the test statistic, if any.
- \texttt{alternative}  a character string describing the alternative hypothesis.
- \texttt{estimates}  the estimates, if any.
- \texttt{null.value}  the estimate under the null hypothesis, if any.

Note

If ties are present, a tie correction is performed and a warning message is given. The GSTTest is sensitive to median differences, likewise to the Siegel-Tukey test. It is thus appropriate to apply this test on the residuals of a one-way ANOVA, rather than on the original data (see example).

References

hartleyTest

See Also

fligner.test, pKruskalWallis, Chisquare, fligner.test

Examples

GSTTest(count ~ spray, data = InsectSprays)

## as means/medians differ, apply the test to residuals
## of one-way ANOVA
ans <- aov(count ~ spray, data = InsectSprays)
GSTTest( residuals( ans) ~ spray, data = InsectSprays)

Description

Performs Hartley’s maximum F-ratio test of the null that variances in each of the groups (samples) are the same.

Usage

hartleyTest(x, ...)

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

## S3 method for class 'formula'
hartleyTest(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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").
Details

If \( x \) is a list, its elements are taken as the samples to be compared for homogeneity of variances. In this case, the elements must all be numeric data vectors, \( g \) is ignored, and one can simply use \( \text{hartleyTest}(x) \) to perform the test. If the samples are not yet contained in a list, use \( \text{hartleyTest}(\text{list}(x, \ldots)) \).

Otherwise, \( x \) must be a numeric data vector, and \( g \) must be a vector or factor object of the same length as \( x \) giving the group for the corresponding elements of \( x \).

Hartley's parametric test requires normality and a nearly balanced design. The p-value of the test is calculated with the function \( \text{pmaxFratio} \) of the package \texttt{SuppDists}.

Value

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

- \textbf{method} a character string indicating what type of test was performed.
- \textbf{data.name} a character string giving the name(s) of the data.
- \textbf{statistic} the estimated quantile of the test statistic.
- \textbf{p.value} the p-value for the test.
- \textbf{parameter} the parameters of the test statistic, if any.
- \textbf{alternative} a character string describing the alternative hypothesis.
- \textbf{estimates} the estimates, if any.
- \textbf{null.value} the estimate under the null hypothesis, if any.

References


See Also

- \texttt{bartlett.test}, \texttt{pmaxFratio}

Examples

- \( \text{hartleyTest}(\text{count} \sim \text{spray}, \text{data} = \text{InsectSprays}) \)
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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset
  an optional vector specifying a subset of observations to be used.

na.action
  a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

Details

The null hypothesis, H₀ : θ₁ = θ₂ = ... = θₖ is tested against a simple order hypothesis, Hₐ : θ₁ ≤ θ₂ ≤ ... ≤ θₖ, θ₁ < θₖ.

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


See Also

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

Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- 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-Terpstra 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-Terpstra Test)*

**Description**

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

**Usage**

```r
jonckheereTest(x, ...)  
```

## Default S3 method:

```r
jonckheereTest(
  x,
  g,
  alternative = c("two.sided", "greater", "less"),
  continuity = FALSE,
  ...
)
```

## S3 method for class 'formula'

```r
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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details
The null hypothesis, H_0 : \theta_1 = \theta_2 = ... = \theta_k is tested against a simple order hypothesis, H_A : \theta_1 \leq \theta_2 \leq ... \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-Terpstra test in the presence of ties was taken from:


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

References

See Also

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

Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- 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-Terpstra 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 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 `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- **subset**: an optional vector specifying a subset of observations to be used.
- **na.action**: a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

### Details
For one-factorial designs with non-normally distributed residuals the Kruskal-Wallis rank sum test can be performed to test the $H_0 : F_1(x) = F_2(x) = \ldots = F_k(x)$ 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


See Also

kruskal.test, pKruskalWallis, Chisquare, FDist

Examples

```r
## 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
g <- factor(x = c(rep(1, length(x)),
                  rep(2, length(y)),
                  rep(3, length(z)));
           labels = c("ns", "oad", "a"))
dat <- data.frame(
g = g,
x = c(x, y, z))

## AD-Test
adKSampleTest(x ~ g, data = dat)

## BWS-Test
bwsKSampleTest(x ~ g, data = dat)

## Kruskal-Test
## Using incomplete beta approximation
kruskalTest(x ~ g, dat, dist="KruskalWallis")
## Using chisquare distribution
kruskalTest(x ~ g, dat, dist="Chisquare")

## Not run:
## Check with kruskal.test from R stats
kruskal.test(x ~ g, dat)

## End(Not run)
## Using Conover's F
kruskalTest(x ~ g, dat, dist="FDist")
```
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

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 p.adjust).

## Not run:
## Check with aov on ranks
anova(aov(rank(x) ~ g, dat))
## Check with oneway.test
oneway.test(rank(x) ~ g, dat, var.equal = TRUE)

## End(Not run)
formula  a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.

data     an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.

subset   an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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


See Also

`Tukey`, `TDist`, `p.adjust`, `kruskalTest`, `kwAllPairsDunnTest`, `kwAllPairsNemenyiTest`
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.
kwAllPairsDunnTest

- **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 `p.adjust`).
- **formula**: A formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
- **data**: An optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- **subset**: An optional vector specifying a subset of observations to be used.
- **na.action**: A function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

**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**


**See Also**

`Normal`, `p.adjust`, `kruskalTest`, `kwAllPairsConoverTest`, `kwAllPairsNemenyiTest`
Examples

```r
## 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

```r
kwAllPairsNemenyiTest(x, ...)  
```

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 distribution for determining the p-value. Defaults to "Tukey".

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

data           
an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset         
an optional vector specifying a subset of observations to be used.

na.action      
a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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 \( m = \frac{k(k - 1)}{2} \) 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

See Also
Tukey, Chisquare, p.adjust, kruskalTest, kwAllPairsDunnTest, kwAllPairsConoverTest
Examples

```r
## 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

```r
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 p.adjust).

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

data  an optional matrix or data frame (or similar: see model.frame) containing
   the variables in the formula formula. By default the variables are taken from
   environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. De-
   faults 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 multi-
variate 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


See Also

`pmvt`, `TDist`, `kruskalTest`, `kwManyOneDunnTest`, `kwManyOneNdwTest`

Examples

```r
## 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`  
*Dunn's Many-to-One Rank Comparison Test*

Description

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

Usage

```
kwManyOneDunnTest(x, ...)
```

## Default S3 method:
kwManyOneDunnTest(
  x,
```
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 p.adjust).

formula

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

data

an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset

an optional vector specifying a subset of observations to be used.

na.action

a function which indicates what should happen when the data contain NAs. Defaults togetOption("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. \( H_i : \theta_0 = \theta_i \), \( (1 \leq i \leq m) \).

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**


**See Also**

`pmvnorm`, `TDist`, `kruskalTest`, `kwManyOneConoverTest`, `kwManyOneNdwTest`

**Examples**

```r
## 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)
```
### 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, ...)  
```

#### 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 `p.adjust`).
- `formula`  
  a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
- `data`  
  an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- `...`  
  further arguments to be passed to or from methods.
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 \( \text{p.adjust.method} = \text{"single-step"} \) 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 \( \text{p.adjust} \).

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 \( \text{kwManyOneDunnTest} \), but there is no tie correction included. Therefore, the implementation of Dunn’s test is superior, when ties are present.

References


See Also

\( \text{pmvt}, \text{TDist}, \text{kruskalTest}, \text{kwManyOneDunnTest}, \text{kwManyOneConoverTest} \)
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)

leTest Testing against Ordered Alternatives (Le's Test)

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 \texttt{model.frame}) containing the variables in the formula \texttt{formula}. By default the variables are taken from \texttt{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 \texttt{getOption("na.action")}.

Details

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

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

Value

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

- \texttt{method} a character string indicating what type of test was performed.
- \texttt{data.name} a character string giving the name(s) of the data.
- \texttt{statistic} the estimated quantile of the test statistic.
- \texttt{p.value} the p-value for the test.
- \texttt{parameter} the parameters of the test statistic, if any.
- \texttt{alternative} a character string describing the alternative hypothesis.
- \texttt{estimates} the estimates, if any.
- \texttt{null.value} the estimate under the null hypothesis, if any.

References


See Also

\texttt{kruskalTest} and \texttt{shirleyWilliamsTest} of the package \texttt{PMCMRplus}, \texttt{kruskal.test} of the library \texttt{stats}.
Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- 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-Terpstra 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

```r
lsdTest(x, ...)
```

---

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

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

## S3 method for class 'aov'
lsdTest(x, ...)
```
Arguments

x  a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.

... 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 model.frame) containing
   the variables in the formula formula. By default the variables are taken from
   environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. De-
  faults to getOption("na.action").

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 \( A_{ij} : \mu_i(x) \neq \mu_j(x), \quad i \neq j \).

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

mackWolfeTest

See Also

TDist, pairwise.t.test

Examples

fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts)
anova(fit)

## also works with fitted objects of class aov
res <- lsddTest(fit)
snmary(res)
snmaryGroup(res)

---

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, \ldots, k\}. Defaults to NULL.

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 `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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.

\[ H_i : \theta_0 = \theta_1 = \cdots = \theta_k \] is tested against the alternative \( A_i : \theta_1 \leq \cdots \leq \theta_p \geq \theta_k \) for some \( p \), with at least one strict inequality.

If \( p = \text{NULL} \) (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 (\( \text{pnorm} \)).

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. \( \text{nperm} = 10000 \) in order to get more precise \( p \)-values. However, this will be on the expense of computational time.

References


Mandel-h

See Also

`pnorm`, `sample`.

Examples

```r
## 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**

```r
qmandelh(p, k, lower.tail = TRUE, log.p = FALSE)
pmandelh(q, k, lower.tail = TRUE, log.p = FALSE)
```

**Arguments**

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

**Value**

`pmandelh` gives the distribution function and `qmandelh` gives the quantile function.

**Source**

The code for `pmandelh` was taken from:
References


See Also

mandelhTest

Examples

```r
## We need a two-sided upper-tail quantile
qmandelh(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

```r
qmandelk(p, k, n, lower.tail = TRUE, log.p = FALSE)
pmandelk(q, k, n, lower.tail = TRUE, log.p = FALSE)
```

Arguments

- `p`: vector of probabilities.
- `k`: number of groups.
- `n`: number of replicates per group.
- `lower.tail`: logical; if TRUE (default), probabilities are P[X <= x] otherwise, P[X > x].
- `log.p`: logical; if TRUE, probabilities are given as log(p).
- `q`: vector of quantiles.

Value

`pmandelk` gives the distribution function and `qmandelk` gives the quantile function.

Source

The code for `pmandelk` was taken from:
Note
The functions are only appropriate for balanced designs.

References

See Also
mandelTest
pmandelh, qmandelh

Examples
qmandelk(0.005, 7, 3, lower.tail=FALSE)

mandelhTest  Mandel’s h Test According to E 691 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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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

See Also
qmandelh pmandelh

Examples
data(Pentosan)
mandelhTest(value ~ lab, data=Pentosan, subset=(material == "A"))

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

- **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 `model.frame`) containing the variables in the formula `formula`. By default, the variables are taken from `environment(formula)`.
- **subset**: An optional vector specifying a subset of observations to be used.
- **na.action**: A function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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


See Also

`qmandelk`, `pmandelk`

Examples

data(Pentosan)
mandelkTest(value ~ lab, data=Pentosan, subset=(material == "A"))
manyOneUTest  

Multiple Comparisons with One Control (U-test)

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 p.adjust)

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

data an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

This function 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 \(p\text{-adjust.method} = \text{"single-step"}\) 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 \(\text{p.adjust}\).

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


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 \(\mu_m - \mu_m \leq \mu_j - \mu_i \leq \mu_l - \mu_i\) 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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset   an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults togetOption("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


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**

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

```r
# 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 `p.adjust`).
- `formula` a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
- `data` an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- `subset` an optional vector specifying a subset of observations to be used.
- `na.action` a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`. 
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 \( \text{p.adjust.method} = \text{"single-step"} \) the Tukey’s studentized range distribution is used to calculate p-values (see \text{Tukey}). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by \text{p.adjust}.

Value

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

- \text{method} \hspace{5mm} \text{a character string indicating what type of test was performed.}
- \text{data.name} \hspace{5mm} \text{a character string giving the name(s) of the data.}
- \text{statistic} \hspace{5mm} \text{lower-triangle matrix of the estimated quantiles of the pairwise test statistics.}
- \text{p.value} \hspace{5mm} \text{lower-triangle matrix of the p-values for the pairwise tests.}
- \text{alternative} \hspace{5mm} \text{a character string describing the alternative hypothesis.}
- \text{p.adjust.method} \hspace{5mm} \text{a character string describing the method for p-value adjustment.}
- \text{model} \hspace{5mm} \text{a data frame of the input data.}
- \text{dist} \hspace{5mm} \text{a string that denotes the test distribution.}

References


See Also

\text{normalScoresTest, normalScoresManyOneTest, normOrder}.
normalScoresManyOneTest

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 p.adjust).
formula       a formula of the form response ~ group where response gives the data values
       and group a vector or factor of the corresponding groups.
data       an optional matrix or data frame (or similar: see model.frame) containing
       the variables in the formula formula. By default the variables are taken from
       environment(formula).
subset       an optional vector specifying a subset of observations to be used.
na.action       a function which indicates what should happen when the data contain NAs. De-
      faults togetOption("na.action").

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


See Also

- `normalScoresTest`, `normalScoresAllPairsTest`, `normOrder`, `pmvt`.

Examples

```r
## 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

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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

For one-factorial designs with non-normally distributed residuals the Lu-Smith normal score test can be performed to test the H₀: F₁(x) = F₂(x) = ... = Fₖ(x) against the Hₐ: Fᵢ(x) ≠ Fⱼ(x) (i ≠ 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**


**See Also**

vanWaerdenTest, kruskalTest, pNormScore

**Examples**

```r
normalScoresTest(count ~ spray, data = InsectSprays)
```

---

**NPMTTest**

*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, \( \theta_m' - \theta_m \leq \theta_j - \theta_i \leq \theta_l' - \theta_l \) \((l \leq i \leq m \text{ and } m' \leq j \leq l')\). 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**

```r
NPMTTest(x, ...)
```

### Default S3 method:

```r
NPMTTest(x, g, ...)
```

### S3 method for class 'formula'

```r
NPMTTest(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.
An optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.

subset

An optional vector specifying a subset of observations to be used.

na.action

A function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

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

- `kwAllPairsNemenyiTest`

Examples

```r
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
      110, 125, 143, 148, 151,
      136, 139, 149, 160, 174)
g <- gl(3,5)
levels(g) <- c("A", "B", "C")
NPMTest(x, g)
```

---

osrtTest

**One-Sided Studentised Range Test**

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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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


Examples

osrtTest(weight ~ group, data = PlantGrowth)
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


See Also

friedmanTest

Examples

```r
## 9 reviewers (blocks)
## assigned ranks to 4 objects (groups).
data(reviewers)
pageTest(reviewers, alternative = "greater")
```

## 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**

plot.mandel

Plotting mandel Objects

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

Plotting PMCMR Objects

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

```r
## data set InsectSprays
ans <- kwAllPairsNemenyiTest(count ~ spray, data = InsectSprays)
plot(ans)
plot(ans, col="red",main="My title", xlab="Spray", "Count")
```

Description

Performs power simulation for one-factorial all-pairs and Many-To-One comparison tests.

Usage

```r
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",
           "tambhaneDunnettTest", "ManyOneUTest", "kwAllPairsNemenyiTest", "kwAllPairsDunnTest",
           "kwAllPairsConoverTest", "normalScoresAllPairsTest", "vanWaerdenAllPairsTest",
           "dscfAllPairsTest", "gamesHowellTest", "lsdTest", "scheffeTest", "tambhaneT2Test",
           "tukeyTest", "dunnettT3Test", "pairwise.t.test", "pairwise.wilcox.test",
           "adManyOneTest", "adAllPairsTest", "bwsManyOneTest", "bwsAllPairsTest",
           "welchManyOneTTest"),
  alternative = c("two.sided", "greater", "less"),
)```
powerMCTests

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 p.adjust).
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 ans 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

```r
## Not run:
mu <- c(0, 0, 1, 2)
n <- c(5, 4, 5, 5)
set.seed(100)
powerMCTests(mu, n, errfn="Normal",
```
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".
**powerOneWayTests**

`powerOneWayTests` function in R is used to calculate power for a one-way ANOVA test. It takes several arguments:

- `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 `kruskalTest`. 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. \( p = \{1, \ldots, k\} \). 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**

```r
## 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

```r
## and powerOneWayTest

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))

## 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
```

## Description

`print.gesdTest` is the `gesdTest` method of the generic `print` function which prints its argument and returns it *invisibly* (via `invisible(x)`).
### Usage

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

### Arguments

- **x**: an object used to select a method.
- **...**: further arguments. Currently ignored.

### Description

`print.mandel` is the `mandel` method of the generic `print` function which prints its argument and returns it invisibly (via `invisible(x)`).

### Usage

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

### Arguments

- **x**: an object used to select a method.
- **...**: further arguments. Currently ignored.

### See Also

- `mandelhTest`, `mandelkTest`

### Description

`print.PMCMR` is the `PMCMR` method of the generic `print` function which prints its argument and returns it invisibly (via `invisible(x)`).

### Usage

```r
## 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**

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

**Arguments**

- `x` an object used to select a method.
- `...` further arguments. Currently ignored.

print.powerPMCMR

**PowerPMCMR Printing**

**Description**

print.powerPMCMR is the `powerPMCMR` method of the generic `print` function which prints its argument and returns it invisibly (via `invisible(x)`).

**Usage**

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

**Arguments**

- `x` an object used to select a method.
- `...` further arguments. Currently ignored.

**See Also**

`powerMCTests`, `powerOneWayTests`
print.steel

Description

print.steel is the *steel* method of the generic `print` function which prints its argument and returns it `invisibly` (via `invisible(x)`).

Usage

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

Arguments

- `x`: an object used to select a method.
- `...`: further arguments. Currently ignored.

print.williams

Description

print.williams is the *williams* method of the generic `print` function which prints its argument and returns it `invisibly` (via `invisible(x)`).

Usage

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

Arguments

- `x`: an object used to select a method.
- `...`: further arguments. Currently ignored.
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_Miner** a numeric vector
- **MAK2** a numeric vector
- **LRE_E100** a numeric vector
- **SPSM** a numeric vector
- **DART** a numeric vector
- **FPLM** a numeric vector
- **LRE_Emax** a numeric vector
- **FKP_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


quadeAllPairsTest

All-Pairs Comparisons for Unreplicated Blocked Data (Quade’s All-Pairs Test)

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
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 "TDist".
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 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"), than 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


See Also

- `quade.test`, `friedmanTest`

Examples

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

## Global test
quade.test(y)

## All-pairs comparisons
quadeAllPairsTest(y, dist="TDist", p.adjust.method="holm")
```


**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**


**References**


**Examples**

```r
data(reviewers)
friedmanTest(reviewers)
pageTest(reviewers)
frdAllPairsExactTest(reviewers, p.adjust = "bonferroni")
```

**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, ...)

## S3 method for class 'aov'
scheffeTest(x, ...)

Arguments

x       a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.
...
    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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset  an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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.
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,
  alternative = c("two.sided", "greater", "less"),
  method = c("look-up", "boot"),
  nperm = 10000,
  ...
)
```r
## S3 method for class 'formula'
shirleyWilliamsTest(
  formula,
  data,
  subset,
  na.action,
  alternative = c("two.sided", "greater", "less"),
  method = c("look-up", "boot"),
  nperm = 10000,
  ...
)
```

### 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`.
- **method**: a character string specifying the test statistic to use. Defaults to "look-up" that uses published Table values of Williams (1972).
- **nperm**: number of permutations for the asymptotic permutation test. Defaults to `1000`. Ignored, if `method = "look-up"`.
- **formula**: a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
- **data**: an optional matrix or data frame (or similar; see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- **subset**: an optional vector specifying a subset of observations to be used.
- **na.action**: a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

### Details

The Shirley-Williams 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:

\[
H_m : \theta_0 = \theta_1 = \ldots = \theta_m, \quad A_m = \theta_0 \leq \theta_1 \leq \ldots \leq \theta_m, \theta_0 < \theta_m \\
H_{m-1} : \theta_0 = \theta_1 = \ldots = \theta_{m-1}, \quad A_{m-1} = \theta_0 \leq \theta_1 \leq \ldots \leq \theta_{m-1}, \theta_0 < \theta_{m-1} \\
\vdots \\
H_1 : \theta_0 = \theta_1, \quad A_1 = \theta_0 < \theta_1
\]
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. This function has included the modifications as recommended by Williams (1986).

If \texttt{method = "look-up"} is selected, the function does not return p-values. Instead the critical t-values as given in the tables of Williams (1972) for \(\alpha = 0.05\) (one-sided) are looked up according to the degree of freedoms \(v = \infty\) and the order number of the dose level \((i)\) and (potentially) modified according to the given extrapolation coefficient \(\beta\).

Non tabulated values are linearly interpolated with the function \texttt{approx}.

For the comparison of the first dose level \((i = 1)\) with the control, the critical z-value from the standard normal distribution is used \(\text{(Normal)}\).

If \texttt{method = "boot"}, the p-values are estimated through an asymptotic boot-strap method. The p-values for \(H_1\) are calculated from the \text{t} distribution with infinite degree of freedom.

\section*{Value}

Either a list with class \"williamsTest\" or a list with class \"PMCMR\".

The list with class \"williamsTest\".

- \texttt{method} a character string indicating what type of test was performed.
- \texttt{data.name} a character string giving the name(s) of the data.
- \texttt{statistic} lower-triangle matrix of the estimated quantiles of the pairwise test statistics.
- \texttt{t.value} lower-triangle matrix of the critical \(t'\)-values for \(\alpha = 0.05\).
- \texttt{df.residual} the degree of freedom
- \texttt{alternative} a character string describing the alternative hypothesis.
- \texttt{model} a data frame of the input data.
- \texttt{dist} a string that denotes the test distribution.

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

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

\section*{Note}

For \texttt{method = "look-up"}, only tests on the level of \(\alpha = 0.05\) can be performed for alternative hypotheses less or greater.

For \texttt{method = "boot"} only the alternative \"two.sided\" can be calculated. One may increase the number of permutations to e.g. \texttt{nperm = 10000} in order to get more precise p-values. However, this will be on the expense of computational time.
References


See Also

`williamsTest`

Examples

## Example from Shirley (1977)
## Reaction times of mice to stimuli to their tails.
y <- c(2.4, 3, 3, 2.2, 2.2, 2.2, 2.2, 2.8, 2, 3, 2.8, 2.2, 3.8, 9.4, 8.4, 3, 3.2, 4.4, 3.2, 7.4, 9.8, 3.2, 5.8, 7.8, 2.6, 2.2, 6.2, 9.4, 7.8, 3.4, 7, 9.8, 9.4, 8.8, 8.8, 3.4, 9, 8.4, 2.4, 7.8)
g <- gl(4, 10)

## Not run:
## two.sided test
summary(shirleyWilliamsTest(y ~ g, method = "boot", alternative = "two.sided"))

## End(Not run)

## one-sided test using look-up table
shirleyWilliamsTest(y ~ g, alternative = "greater")

---

`siegelTukeyTest` *Siegel-Tukey Rank Dispersion Test*

Description

Performs Siegel-Tukey non-parametric rank dispersion test.

Usage

`siegelTukeyTest(x, ...)`

## Default S3 method:
`siegelTukeyTest(`

  `x,`

  `y,`

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

  `median.corr = FALSE,`
...}

## S3 method for class 'formula'

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

### Arguments

- **x, y**
  - numeric vectors of data values.
- **...**
  - further arguments to be passed to or from methods.
- **alternative**
  - a character string specifying the alternative hypothesis, must be one of "two.sided" (default), "greater" or "less". You can specify just the initial letter.
- **median.corr**
  - logical indicator, whether median correction should be performed prior testing. 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 `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
- **subset**
  - an optional vector specifying a subset of observations to be used.
- **na.action**
  - a function which indicates what should happen when the data contain NAs. Defaults to `getOption("na.action")`.

### Details

Let \( x \) and \( y \) denote two identically and independently distributed variables of at least ordinal scale. Further, let \( \theta \) and \( \lambda \) denote location and scale parameter of the common, but unknown distribution. Then for the two-tailed case, the null hypothesis \( H: \lambda_x/\lambda_y = 1 | \theta_x = \theta_y \) is tested against the alternative, \( A: \lambda_x/\lambda_y \neq 1 \).

The data are combinedly ranked according to Siegel-Tukey. The ranking is done by alternate extremes (rank 1 is lowest, 2 and 3 are the two highest, 4 and 5 are the two next lowest, etc.). If no ties are present, the p-values are computed from the Wilcoxon distribution (see `Wilcoxon`). In the case of ties, a tie correction is done according to Sachs (1997) and approximate p-values are computed from the standard normal distribution (see `Normal`). If both medians differ, one can correct for medians to increase the specificity of the test.

### 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

References

Examples
## Sachs, 1997, p. 376
A <- c(10.1, 7.3, 12.6, 2.4, 6.1, 8.5, 8.8, 9.4, 10.1, 9.8)
B <- c(15.3, 3.6, 16.5, 2.9, 3.3, 4.2, 4.9, 7.3, 11.7, 13.7)
siegelTukeyTest(A, B)

## from example var.test
x <- rnorm(50, mean = 0, sd = 2)
y <- rnorm(30, mean = 1, sd = 1)
siegelTukeyTest(x, y, median.corr = TRUE)

## directional hypothesis
A <- c(33, 62, 84, 85, 88, 93, 97)
B <- c(4, 16, 48, 51, 66, 98)
siegelTukeyTest(A, B, alternative = "greater")

skillingsMackTest skillingsMackTest

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


See Also

friedmanTest, durbinTest

Examples

```r
## 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)

---

### snkTest

**Student-Newman-Keuls Test**

**Description**

Performs Student-Newman-Keuls all-pairs comparisons test for normally distributed data with equal group variances.

**Usage**

```r
snkTest(x, ...)
```

**Arguments**

- **x**  
a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an `aov` fit.
- **...**  
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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details
For all-pairs comparisons in an one-factorial layout with normally distributed residuals and equal variances Student-Newman-Keuls 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
Student (1927) Errors of routine analysis, Biometrika 19, 151–164.

See Also
Tukey, TukeyHSD tukeyTest
Examples

```r
fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts)
anova(fit)

## also works with fitted objects of class aov
res <- snkTest(fit)
sn澳大 (res)
sn澳门 (res)
```

---

`spearmanTest` Testing against Ordered Alternatives (Spearman Test)

Description

Performs a Spearman type test for testing against ordered alternatives.

Usage

```r
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 `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
spearmanTest

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

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

Examples
## Example from Sachs (1997, p. 402)
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-Terpstra test

`jonckheereTest(x, g)`

### Le's test

`leTest(x, g)`

### Spearman type test

`spearmanTest(x, g)`

### Murakami's BWS trend test

`bwsTrendTest(x, g)`

---

#### `steelTest`

**Steel's Many-to-One Rank Test**

**Description**

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

**Usage**

```r
steelTest(x, ...)  
```

**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`
For many-to-one comparisons (pairwise comparisons with one control) in an one-factorial balanced layout with non-normally distributed residuals Steels's non-parametric single-step test can be performed. Let there be \( k \) treatment levels (excluding the control), then \( k \) pairwise comparisons can be performed between the \( i \)-th treatment level and the control. \( H_i : \theta_0 = \theta_i \) is tested in the one-tailed case (less) against \( A_i : \theta_0 > \theta_i \), \((1 \leq i \leq k)\).

For each control - treatment level the data are ranked in increasing order. The ranksum \( R_i \) for the \( i \)-th treatment level is compared to a critical \( R \) value and is significantly \((p = 0.05)\) less, if \( R_i \leq R \). For the alternative = "greater" the sign is changed.

The function does not return p-values. Instead the critical \( R \)-values as given in the tables of USEPA (2002) for \( \alpha = 0.05 \) (one-sided, less) are looked up according to the balanced sample sizes \((n)\) and the order number of the dose level \((i)\).

**Value**

A list with class "steelTest" 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 ranksum for the \( i \)-th treatment level
- **R.crit** lower-triangle matrix of critical \( R \)-values for \( \alpha = 0.05 \).
- **alternative** a character string describing the alternative hypothesis.
- **model** a data frame of the input data.
- **dist** a string that denotes the test distribution.

There are print and summary methods available.

**Source**

The critical rank sum values were taken from Table E.5 of USEPA (2002).


**Note**

Steel's Many-to-One Rank test is only applicable for balanced designs and directional hypotheses. An error message will occur, if the design is unbalanced. In the current implementation, only one-sided tests on the level of \( \alpha = 0.05 \) can be performed.
References


See Also

`wilcox.test`, `pairwise.wilcox.test`, `manyOneUTest`, `shirleyWilliamsTest`, `kwManyOneDunnTest`, `kwManyOneNdwTest`, `kwManyOneConoverTest`, `print.steel`, `summary.steel`

Examples

```r
## Example from Sachs (1997, p. 402)
g <- gl(3, 5)
levels(g) <- c("0", "I", "II")

## Steel's Test
steelTest(x ~ g)

## Example from USEPA (2002):
## Reproduction data from a Ceriodaphnia dubia
## 7-day chronic test to several concentrations
## of effluent. Dose level 50% is excluded.
x <- c(20, 26, 26, 23, 24, 27, 26, 23, 27, 24, 13, 15, 14, 13, 18, 22, 18, 22, 13, 13, 23, 22, 14, 22, 20, 23, 20, 23, 25, 24, 25, 21, 9, 0, 9, 7, 6, 10, 12, 14, 9, 13, rep(0,10))
g <- gl(6, 10)
levels(g) <- c("Control", "3%", "6%", "12%", "25%", "50%")

## NOEC at 3%, LOEC at 6%
steelTest(x ~ g, subset = g != "50%", alternative = "less")
```

**summary.gesdTest**

*Summarize an gesdTest Object*

**Description**

Summarize an object of class `gesdTest`. 
Usage

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

Arguments

- `object`: an object of class "gesdTest".
- `...`: further arguments. Currently ignored.

Description

`summary.mandel` is a function used to produce result summaries of the results of the functions `mandelhTest` or `mandelkTest`.

Usage

```r
## 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`

Description

`summary.PMCMR` is a function to summarize an object of class `PMCMR`.

Usage

```r
## 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)

summary.steel  Summarize a steel Object

Description

  Summarize an object of class steel.

Usage

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

Arguments

  object  an object of class "steel".
  ...  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.steel, summaryGroup.

Examples

  ans <- vanWaerdenAllPairsTest(count ~ spray, InsectSprays)
  summary(ans)
**summary.williams**  
*Summarize an williams Object*

**Description**

Summarize an object of class *williams*.

**Usage**

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

**Arguments**

- `object` an object of class "williams".
- `...` 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.williams`, `summaryGroup`.

**Examples**

```r
ans <- vanWaerdenAllPairsTest(count ~ spray, InsectSprays)
summary(ans)
```

**summaryGroup**  
*Grouped Summary of an PMCMR Object*

**Description**

Performes a grouped summary on an PMCMR object.

**Usage**

```r
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

tamhaneDunnettTest  Tamhane-Dunnett Many-to-One Comparison Test

Description

Performs Tamhane-Dunnett’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-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.

Usage

tamhaneDunnettTest(x, ...)

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

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

## S3 method for class 'aov'
tamhaneDunnettTest(x, alternative = c("two.sided", "greater", "less"), ...)

Arguments

x  a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.

...  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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults togetOption("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

See Also
pmvt, welchManyOneTTest

Examples

```r
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)
## works with object of class aov
summary(tamhaneDunnettTest(fit, alternative = "greater"))
```
tamhaneT2Test  

Tamhane’s T2 Test

Description
Performs Tamhane’s T2 (or T2’) all-pairs comparison test for normally distributed data with unequal variances.

Usage

tamhaneT2Test(x, ...)

## Default S3 method:
tamhaneT2Test(x, g, welch = TRUE, ...)

## S3 method for class 'formula'
tamhaneT2Test(formula, data, subset, na.action, welch = TRUE, ...)

## S3 method for class 'aov'
tamhaneT2Test(x, welch = TRUE, ...)

Arguments

x  
a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.

...  
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.

welch  
indicates, whether Welch’s approximate solution for calculating the degree of freedom shall be used or, as usually, \( df = N - 2 \). Defaults to TRUE.

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

data  
an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  
an optional vector specifying a subset of observations to be used.

na.action  
a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

Details
For all-pairs comparisons in an one-factorial layout with normally distributed residuals but unequal groups variances the T2 test (or 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 \( A_{ij} : \mu_i(x) \neq \mu_j(x), \; i \neq j \).
T2 test uses Welch’s approximate solution for calculating the degree of freedom. T2’ test uses the usual $df = N - 2$ approximation. A warning message appears in the modified T2’ test, if none of in Tamhane (1979) given conditions for nearly balanced sample sizes and nearly balanced standard errors is true.

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.

**Note**

T2 test is basically an all-pairs pairwise-t-test. Similar results can be obtained with `pairwise.t.test(..., var.equal=FALSE, p.adjust.method = FALSE)`.

Thanks to Sirio Bolaños for his kind suggestion for adding T2’ test into this function.

**References**


**See Also**

- `dunnettT3Test`

**Examples**

```r
fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts) # var1 = varN
anova(fit)

## also works with fitted objects of class aov
res <- tamhaneT2Test(fit)
summary(res)
summaryGroup(res)
res

## compare with pairwise.t.test
```
trout <- pairwise.t.test(chickwts$weight, chickwts$feed, pool.sd = FALSE, p.adjust.method = "none")
p.adj.sidak <- function(p, m) sapply(p, function(p) min(1, 1 - (1 - p)^m))
p.raw <- as.vector(WT$p.value)
m <- length(p.raw[!is.na(p.raw)])
PADJ <- matrix(ans <- p.adj.sidak(p.raw, m), nrow = 5, ncol = 5)
colnames(PADJ) <- colnames(WT$p.value)
rownames(PADJ) <- rownames(WT$p.value)
PADJ

## same without Welch's approximate solution
summary(T2b <- tamhaneT2Test(fit, welch = FALSE))

data.frame(chickwts)

### 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**


**References**

Tukey's Multiple Comparison Test

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, ...)
  ## S3 method for class 'aov'
tukeyTest(x, ...)

Arguments

x          a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.
...        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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset     an optional vector specifying a subset of observations to be used.
na.action  a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

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 \( m = \frac{k(k - 1)}{2} \) 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


See Also

Tukey, TukeyHSD

Examples

```r
fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts)
anova(fit)

## also works with fitted objects of class aov
res <- tukeyTest(fit)
summary(res)
summaryGroup(res)
```

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,
    ...
)

## S3 method for class 'aov'
uryWigginsHochbergTest(x, p.adjust.method = p.adjust.methods, ...)

Arguments

x       a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.
...
    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 p.adjust).
formula a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data    an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset  an optional vector specifying a subset of observations to be used.
na.action    a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

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


See Also

dunnettT3Test

Examples

```r
fit <- aov(weight ~ feed, chickwts)
shapiro.test(residuals(fit))
bartlett.test(weight ~ feed, chickwts) # var1 = varN
anova(fit)
## also works with fitted objects of class aov
res <- uryWigginsHochbergTest(fit)
summary(res)
summaryGroup(res)
```
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

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 p.adjust).
formula  a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data     an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset   an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").
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 \( \text{p.adjust.method} = "\text{single-step}" \) the Tukey’s studentized range distribution is used to calculate p-values (see \text{Tukey}). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by \text{p.adjust}.

Value

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

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

References


See Also

\text{vanWaerdenTest}, \text{vanWaerdenManyOneTest}, \text{normOrder}.
vanWaerdenManyOneTest

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.

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

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

data an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset an optional vector specifying a subset of observations to be used.

na.action a function which indicates what should happen when the data contain NAs. Defaults togetOption("na.action").

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


See Also

vanWaerdenTest, vanWaerdenAllPairsTest, pmvt.

Examples

```r
## 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  

vanWaerdenTest

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 model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).

subset  an optional vector specifying a subset of observations to be used.

na.action  a function which indicates what should happen when the data contain NAs. Defaults togetOption("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 $H_0: F_1(x) = F_2(x) = \ldots = F_k(x)$ 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


See Also

kruskalTest, normalScoresTest

Examples

vanWaerdenTest(count ~ spray, data = InsectSprays)

---

**welchManyOneTTest**

*Welch's Many-To-One Comparison Test*

**Description**

Performs Welch's t-test for multiple comparisons with one control.

**Usage**

```r
welchManyOneTTest(x, ...)
```

### Default S3 method:

```r
welchManyOneTTest(
  x,
  g, 
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = p.adjust.methods,
  ...
)
```

### S3 method for class 'formula'

```r
welchManyOneTTest(
  formula, 
  data, 
  subset, 
  na.action, 
  alternative = c("two.sided", "greater", "less"), 
  p.adjust.method = p.adjust.methods, 
  ...
)
```

### S3 method for class 'aov'

```r
welchManyOneTTest(
  formula, 
  data, 
  subset, 
  na.action, 
  alternative = c("two.sided", "greater", "less"), 
  p.adjust.method = p.adjust.methods, 
  ...
)
```
welchManyOneTTest(
  x,
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = p.adjust.methods,
  ...
)

Arguments

x       a numeric vector of data values, a list of numeric data vectors or a fitted model object, usually an aov fit.
...     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 p.adjust).
formula a formula of the form response ~ group where response gives the data values and group a vector or factor of the corresponding groups.
data an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula formula. By default the variables are taken from environment(formula).
subset an optional vector specifying a subset of observations to be used.
na.action a function which indicates what should happen when the data contain NAs. Defaults to getOption("na.action").

Details

For many-to-one comparisons in an one-factorial layout with normally distributed residuals and unequal variances Welch’s t-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 $A_i : \mu_0(x) \neq \mu_i(x), \ 1 \leq i \leq k - 1$.

This function is basically a wrapper function for t.test(..., var.equal = FALSE). The p-values for the test are calculated from the t distribution and can be adjusted with any method that is implemented in p.adjust.methods.

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

Welch, B. L. (1947) The generalization of "Student's" problem when several different population variances are involved, *Biometrika* 34, 28–35.


See Also

`pairwise.t.test, t.test, p.adjust, tamhaneDunnettTest`

Examples

```r
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)
anova(fit)
summary(welchManyOneTTest(fit, alternative = "greater", p.adjust="holm"))
```

---

**williamsTest**  

**Williams Trend Test**

**Description**

Performs Williams' test for contrasting increasing (decreasing) dose levels of a treatment.

**Usage**

```r
williamsTest(x, 

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

## S3 method for class 'formula'
williamsTest(
  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 model.frame) containing
the variables in the formula formula. By default the variables are taken from
environment(formula).
subset          an optional vector specifying a subset of observations to be used.
na.action       a function which indicates what should happen when the data contain NAs. De-
defaults to getOption("na.action").

Details

Williams’ test is a step-down trend test for testing several treatment levels with a zero control in
a one-factorial design with normally distributed errors of homogeneous variance. Let there be k
groups including the control and let the zero dose level be indicated with i = 0 and the treatment
levels indicated as 1 ≤ i ≤ m, then the following m = k − 1 hypotheses are tested:

H_m : \bar{x}_0 = m_1 = \ldots = m_m, \quad A_m : \bar{x}_0 \geq m_1 \geq \ldots \geq m_m, \bar{x}_0 < m_m
H_{m-1} : \bar{x}_0 = m_1 = \ldots = m_{m-1}, \quad A_{m-1} : \bar{x}_0 \geq m_1 \geq \ldots \geq m_{m-1}, \bar{x}_0 < m_{m-1}
\vdots
H_1 : \bar{x}_0 = m_1, \quad A_1 : \bar{x}_0 < m_1,

where m_i denotes the isotonic mean of the ith dose level group. 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 function does not return p-values. Instead the critical t-values as given in the tables of Williams
(1972) for \alpha = 0.05 (one-sided) are looked up according to the degree of freedoms (v) and the
order number of the dose level (i) and (potentially) modified according to the given extrapolation
coefficient \beta.
Non tabulated values are linearly interpolated as recommended by Williams (1972). The function
approx is used.
For the comparison of the first dose level (i = 1) with the control, the critical t-value from the Student
t distribution is used (TDist).
Value

A list with class "williamsTest" 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.
- **t.value**: lower-triangle matrix of the critical $t$'-values for $\alpha = 0.05$.
- **df.residual**: the degree of freedom.
- **alternative**: a character string describing the alternative hypothesis.
- **model**: a data frame of the input data.
- **dist**: a string that denotes the test distribution.

There are print and summary methods available.

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**:


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


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.

Note

In the current implementation, only tests on the level of $\alpha = 0.05$ can be performed. The included extrapolation function assumes either a balanced design, or designs, where the number of replicates in the control exceeds the number of replicates in the treatment levels. A warning message appears, if the following condition is not met, $1 \leq n_0/n_i \leq 6$ for $1 \leq i \leq m$.

References

Williams, D. A. (1971) A test for differences between treatment means when several dose levels are compared with a zero dose control, Biometrics 27, 103–117.

Williams, D. A. (1972) The comparison of several dose levels with a zero dose control, Biometrics 28, 519–531.

See Also

TDist, approx, print.williams, summary.williams
## Example from Sachs (1997, p. 402)

```r
g <- gl(3,5)
levels(g) <- c("0", "I", "II")

## Williams Test
williamsTest(x ~ g)
```
Index

* datasets
  algae, 10
  Pentosan, 116
  qPCR, 126
  reviewers, 129
  trout, 152
* distribution
  Cochran, 25
  Dgrubbs, 30
  Grubbs, 67
  Mandel-h, 99
  Mandel-k, 100
* friedmanranks
  durbinAllPairsTest, 39
  durbinTest, 41
  frdAllPairsConoverTest, 42
  frdAllPairsExactTest, 44
  frdAllPairsMillerTest, 47
  frdAllPairsNemenyiTest, 49
  frdAllPairsSiegelTest, 52
  frdManyOneDemsarTest, 54
  frdManyOneExactTest, 56
  frdManyOneNemenyiTest, 58
* htest
  adAllPairsTest, 4
  adKSampleTest, 6
  adManyOneTest, 8
  bwsAllPairsTest, 12
  bwsKSampleTest, 14
  bwsManyOneTest, 16
  bwsTrendTest, 18
  chaAllPairsNashimotoTest, 21
  chackoTest, 23
  coxchranTest, 26
  cuzickTest, 28
  doubleGrubbsTest, 31
  dscfAllPairsTest, 32
  duncanTest, 34
  dunnettT3Test, 35
  dunnettTest, 37
  durbinAllPairsTest, 39
  durbinTest, 41
  frdAllPairsConoverTest, 42
  frdAllPairsExactTest, 44
  frdAllPairsMillerTest, 47
  frdAllPairsNemenyiTest, 49
  frdAllPairsSiegelTest, 52
  frdManyOneDemsarTest, 54
  frdManyOneExactTest, 56
  frdManyOneNemenyiTest, 58
  friedmanTest, 60
  gamesHowellTest, 63
  gesdTest, 64
  goreTest, 65
  grubbsTest, 68
  GSTTest, 69
  hartleyTest, 71
  johnsonTest, 73
  jonckheereTest, 75
  kruskalTest, 77
  kwAllPairsConoverTest, 80
  kwAllPairsDunnTest, 82
  kwAllPairsNemenyiTest, 84
  kwManyOneConoverTest, 86
  kwManyOneDunnTest, 88
  kwManyOneNdwTest, 91
  leTest, 93
  lsdTest, 95
  mackWolfeTest, 97
  mandelhTest, 101
  mandelkTest, 102
  manyOneUTest, 104
  MTest, 105
  normalScoresAllPairsTest, 107
  normalScoresManyOneTest, 108
  normalScoresTest, 110
  NPMTest, 112
  osrtTest, 113
INDEX

pageTest, 115
quadeAllPairsTest, 127
scheffeTest, 129
shirleyWilliamsTest, 131
siegelTukeyTest, 134
skillingsMackTest, 136
snkTest, 138
spearmanTest, 140
steelTest, 142
tamhaneDunnettTest, 148
tamhaneT2Test, 150
tukeyTest, 153
uryWigginsHochbergTest, 154
vanWaerdenAllPairsTest, 157
vanWaerdenManyOneTest, 158
vanWaerdenTest, 161
welchManyOneTTest, 162
williamsTest, 164

* kruskalranks
  kruskalTest, 77
  kwAllPairsConoverTest, 80
  kwAllPairsDunnTest, 82
  kwAllPairsNemenyiTest, 84
  kwManyOneConoverTest, 86
  kwManyOneDunnTest, 88
  kwManyOneNdwTest, 91
  mackWolfeTest, 97

* methods
  summary.gesdTest, 144
  summary.mandel, 145
  summary.PMCMR, 145
  summary.steel, 146
  summary.williams, 147
  summaryGroup, 147

* misc
  powerMCTests, 118
  powerOneWayTests, 120

* nonparametric
  adKSampleTest, 6

* nonparametric
  adAllPairsTest, 4
  adManyOneTest, 8
  bwsAllPairsTest, 12
  bwsKSampleTest, 14
  bwsManyOneTest, 16
  bwsTrendTest, 18
  chaAllPairsNashimotoTest, 21
  chackoTest, 23
cuzickTest, 28
dscfAllPairsTest, 32
durbinAllPairsTest, 39
durbinTest, 41
frdAllPairsConoverTest, 42
frdAllPairsExactTest, 44
frdAllPairsMillerTest, 47
frdAllPairsNemenyiTest, 49
frdAllPairsSiegelTest, 52
frdManyOneDemsarTest, 54
frdManyOneExactTest, 56
frdManyOneNemenyiTest, 58
friedmanTest, 60
goreTest, 65
GSTTest, 69
johnsonTest, 73
jonckheereTest, 75
kwAllPairsConoverTest, 80
kwAllPairsDunnTest, 82
kwAllPairsNemenyiTest, 84
kwManyOneConoverTest, 86
kwManyOneDunnTest, 88
kwManyOneNdwTest, 91
leTest, 93
mackWolfeTest, 97
manyOneUTest, 104
normalScoresAllPairsTest, 107
normalScoresManyOneTest, 108
normalScoresTest, 110
NPMTest, 112
pageTest, 115
quadeAllPairsTest, 127
shirleyWilliamsTest, 131
siegelTukeyTest, 134
skillingsMackTest, 136
spearmanTest, 140
vanWaerdenAllPairsTest, 157
vanWaerdenManyOneTest, 158
vanWaerdenTest, 161

* normalscores
  normalScoresAllPairsTest, 107
  normalScoresManyOneTest, 108
  normalScoresTest, 110
  NPMTest, 112
  pageTest, 115
  quadeAllPairsTest, 127
  shirleyWilliamsTest, 131
  siegelTukeyTest, 134
  skillingsMackTest, 136
  spearmanTest, 140
  vanWaerdenAllPairsTest, 157
  vanWaerdenManyOneTest, 158
  vanWaerdenTest, 161

* outliers
  cochranTest, 26
doubleGrubbsTest, 31
gesdTest, 64
grubbsTest, 68
  * parametric
duncanTest, 34
dunnettT3Test, 35
dunnettTest, 37
gamesHowellTest, 63
hartleyTest, 71
lsdTest, 95
MTest, 105
scheffeTest, 129
snkTest, 138
tamhaneDunnettTest, 148
tamhaneT2Test, 150
tukeyTest, 153
uryWigginsHochbergTest, 154
welchManyOneTTest, 162
  * plot
barPlot, 11
plot.PMCMR, 117
  * print
print.gesdTest, 122
print.mandel, 123
print.PMCMR, 123
print.powerOneWayPMCMR, 124
print.powerPMCMR, 124
print.steel, 125
print.williams, 125
  * testpower
powerMCTests, 118
powerOneWayTests, 120
  * trendtest
bwsTrendTest, 18
chackoTest, 23
cuzickTest, 28
johnsonTest, 73
jonckheereTest, 75
leTest, 93
shirleyWilliamsTest, 131
spearmanTest, 140
  * umbrellatess
mackWolfeTest, 97
  * univariate
gesdTest, 64
grubbsTest, 68
  * wilcoxonranks
cuzickTest, 28
dscfAllPairsTest, 32
manyOneUTest, 104
ad.pval, 6, 7, 10
adAllPairsTest, 4, 7, 10
adKSampleTest, 6, 6, 10
adManyOneTest, 6, 7, 8
algae, 10
aov, 34, 36, 38, 63, 96, 130, 138, 148, 150, 153, 155, 163
approx, 133, 165, 166
barPlot, 11
bartlett.test, 27, 72
boxplot, 118
bws_cdf, 17, 18
bws_stat, 17, 18
bws_test, 13
bwsAllPairsTest, 12, 15, 20
bwsKSampleTest, 14
bwsManyOneTest, 15, 16, 20
bwsTrendTest, 18
chaAllPairsNashimotoTest, 21
chackoTest, 21, 22, 23
Chisquare, 71, 79, 85
Cochran, 25
cochranTest, 26
cuzickTest, 28
default (kwAllPairsDunnTest), 82
Dgrubbs, 30
doubleGrubbsTest, 31
dscfAllPairsTest, 32
duncanTest, 34
dunnettT3Test, 35, 151, 156
dunnettTest, 37
durbinAllPairsTest, 39
durbinTest, 40, 41, 66, 137
FDist, 26, 79, 131
fligner.test, 27, 70, 71
frdAllPairsConoverTest, 42, 46, 48, 51, 53
frdAllPairsExactTest, 43, 44, 48, 51, 53
frdAllPairsMillerTest, 43, 46, 47, 51, 53
frdAllPairsNemenyiTest, 43, 46, 48, 49, 53
frdAllPairsSiegelTest, 43, 46, 48, 51, 52
frdManyOneDemsarTest, 54, 57, 60
frdManyOneExactTest, 56, 56, 60
frdManyOneNemenyiTest, 56, 56, 58
friedman.test, 43, 46, 48, 51, 53, 56, 57, 60, 61
INDEX 171

friedmanTest, 43, 46, 48, 51, 53, 56, 57, 60, 66, 116, 128, 137

gamesHowellTest, 63
gesdTest, 64
goreTest, 65
Grubbs, 31, 67
grubbsTest, 68
GSTTest, 69

hartleyTest, 71

invisibile, 122–125

johnsonTest, 73
jonckheereTest, 75

kruskal.test, 20, 24, 29, 74, 77, 79, 94, 141
kruskalTest, 20, 24, 29, 74, 77, 81, 83, 85, 88, 90, 92, 94, 112, 121, 141, 162
kwAllPairsConoverTest, 80, 83, 85
kwAllPairsDunnTest, 81, 82, 85
kwAllPairsNemenyiTest, 81, 83, 84, 112, 113
kwManyOneConoverTest, 86, 90, 92, 144
kwManyOneDunnTest, 88, 88, 92, 144
kwManyOneNdwTest, 88, 90, 91, 144

leTest, 93
lsdTest, 95

mackWolfeTest, 97
Mandel-h, 99
Mandel-k, 100
mandelhTest, 100, 101, 123, 145
mandelkTest, 101, 102, 123, 145
manyOneUTest, 104, 144
model.frame, 5, 7, 9, 12, 14, 17, 19, 22, 23, 27, 29, 33, 34, 36, 38, 50, 63, 70, 71, 73, 76, 78, 81, 83, 85, 87, 89, 91, 94, 96, 98, 101, 103, 104, 106, 107, 109, 111, 113, 114, 130, 132, 135, 139, 140, 143, 149, 150, 153, 155, 157, 159, 161, 163, 165

MTest, 105
murakami_cdf, 17, 18
murakami_stat, 17, 18

Normal, 22, 83, 105, 133, 135
normalScoresAllPairsTest, 107, 110

normalScoresManyOneTest, 108, 108
normalScoresTest, 108, 110, 110, 162
normOrder, 108, 110, 158
NPMTest, 112

osrtTest, 105, 113

p.adjust, 5, 9, 12, 13, 17, 39, 43, 45, 52, 53, 55, 57, 80, 81, 83, 85, 87, 89, 91, 92, 104, 105, 107–110, 119, 127, 155, 157–160, 163, 164
p.adjust.methods, 22, 40, 127, 163
pageTest, 115
pairwise.t.test, 96, 97, 164
pairwise.wilcox.test, 34, 144
pcochran, 27
pcochran (Cochran), 25
pdgrubbs, 31
pdgrubbs (Dgrubbs), 30
Pentosan, 116
pgrubbs, 68
pgrubbs (Grubbs), 67
pKruskalWallis, 70, 71, 78, 79
plot.mandel, 117
plot.PMCMR, 117
pmandelh, 101, 102
pmandelh (Mandel-h), 99
pmandelk, 103
pmandelk (Mandel-k), 100
pmaxFratio, 72
pmvnorm, 59, 60, 90, 105
pmvt, 36–39, 88, 92, 110, 148, 149, 160
pnorm, 98, 99
pNormScore, 111, 112
power.anova.test, 121
power.MCTests, 118, 121, 124
powerOneWayTests, 120, 124
print, 122–125
print (print.mandel), 123
print.gesdTest, 122
print.mandel, 123
print.PMCMR, 123, 146
print.powerOneWayPMCMR, 124
print.powerPMCMR, 124
print.steel, 125, 144, 146
print.williams, 125, 147, 166
ptukey, 64
pwr.anova.test, 121
qcochran (Cochran), 25
qgrubbs (Grubbs), 67
qmandelh, 101, 102
qmandelh (Mandel-h), 99
qmandelk, 103
qmandelk (Mandel-k), 100
qPCR, 126
quade.test, 127, 128
quadeAllPairsTest, 127
reviewers, 129
sample, 15, 19, 20, 99
scheffeTest, 129
set.seed, 60
shirleyWilliamsTest, 20, 24, 29, 74, 77, 94, 131, 141, 144
siegelTukeyTest, 134
skillingsMackTest, 66, 136
snkTest, 138
spearmanTest, 140
steelTest, 142
summary.gesdTest, 144
summary.mandel, 145
summary.PMCMR, 145, 148
summary.steel, 144, 146
summary.williams, 147, 166
summaryGroup, 146, 147, 147
t.test, 163, 164
tamhaneDunnettTest, 148, 164
tamhaneT2Test, 150
TDist, 67, 81, 88, 90, 92, 97, 165, 166
tout, 152
Tukey, 22, 34, 35, 81, 85, 108, 139, 154, 158
TukeyHSD, 35, 139, 154
tukeyTest, 35, 131, 139, 153
uryWigginsHochbergTest, 154
vanWaerdenAllPairsTest, 157, 160
vanWaerdenManyOneTest, 158, 158
vanWaerdenTest, 112, 158, 160, 161
vanWeardenTest.default
(vanWaerdenTest), 161
welchManyOneTTest, 149, 162
wilcox.test, 105, 144
Wilcoxon, 135
williamsTest, 134, 164