Package: PMCMRplus (via r-universe)

October 9, 2024

Type Package

Title Calculate Pairwise Multiple Comparisons of Mean Rank Sums
Extended

Version 1.9.12

Date 2024-09-07

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 (>= 3.5.0)

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

Suggests xtable, graphics, knitr, rmarkdown, car, e1071, multcomp, pwr, NSM3

SystemRequirements gmp (>= 4.2.3), mpfr (>= 3.0.0) | 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 7.3.1

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 2024-09-08 10:10:03 UTC

Contents

adAllPairsTest 5
adKSampleTest 7
adManyOneTest 9
algae 11
barPlot 12
bwsAllPairsTest 13
bwsKSampleTest 15
bwsManyOneTest 18
bwsTrendTest 20
chaAllPairsNashimotoTest 23
chackoTest 26
chenJanTest 29
chenTest 31
Cochran 34
cochranTest 35
cuzickTest 37
Dgrubbs 39
doubleGrubbsTest 40
dscfAllPairsTest 42
duncanTest 43
dunnettT3Test 45

dunnettTest 47
durbinAllPairsTest 49
durbinTest 51
flignerWolfeTest 52
frdAllPairsConoverTest 55
frdAllPairsExactTest 58
frdAllPairsMillerTest 60
frdAllPairsNemenyiTest 62
frdAllPairsSiegelTest 65
frdHouseTest 67
frdManyOneDemsarTest 70
frdManyOneExactTest 72
frdManyOneNemenyiTest 74
friedmanTest 76
gamesHowellTest 78
gesdTest 80
goreTest 81
Grubbs 83
grubbsTest 84
GSTTest 85
hartleyTest 87
hayterStoneTest 89
hsAllPairsTest 92
johnsonTest 95
jonckheereTest 97
kruskalTest 100
kwAllPairsConoverTest 103
kwAllPairsDunnTest 105
kwAllPairsNemenyiTest 108
kwManyOneConoverTest 110
kwManyOneDunnTest 113
kwManyOneNdwTest 116
leTest 118
lsdTest 121
mackWolfeTest 123
Mandel-h 125
Mandel-k 126
mandelhTest 127
mandelkTest 128
manyOneUTest 130
medianAllPairsTest 132
medianTest 134
mrrTest 137
MTest 139
normalScoresAllPairsTest 141
normalScoresManyOneTest 143
normalScoresTest 145
NPMTest 147

osrtTest 150
pageTest 153
Pentosan 154
plot.mandel 155
plot.PMCMR 155
power.dunnett.test 156
power.tukey.test 158
power.williams.test 159
powerMCTests 161
powerOneWayTests 162
print.gesdTest 165
print.mandel 165
print.osrt 166
print.PMCMR 167
print.powerOneWayPMCMR 167
print.powerPMCMR 168
print.trendPMCMR 168
qDunnett 169
qPCR 170
quadeAllPairsTest 171
reviewers 173
scheffeTest 174
shanTest 176
shirleyWilliamsTest 178
siegelTukeyTest 182
skillingsMackTest 184
snkTest 186
spearmanTest 188
steelsKSampleTest 191
steelTest 193
stepDownTrendTest 195
summary.gesdTest 198
summary.mandel 198
summary.osrt 199
summary.PMCMR 199
summary.trendPMCMR 200
summaryGroup 200
tamhaneDunnettTest 201
tamhaneT2Test 203
toTidy 206
trout 206
tukeyTest 207
uryWigginsHochbergTest 209
vanWaerdenAllPairsTest 211
vanWaerdenManyOneTest 213
vanWaerdenTest 215
welchManyOneTTest 216
williamsTest 218

Index 222

---

adAllPairsTest *Anderson-Darling All-Pairs Comparison Test*

Description

Performs Anderson-Darling all-pairs comparison test.

Usage

```
adAllPairsTest(x, ...)

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

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

Arguments

`x`	a numeric vector of data values, or a list of numeric data vectors.
`...`	further arguments to be passed to or from methods.
`g`	a vector or factor object giving the group for the corresponding elements of `"x"`. Ignored with a warning if `"x"` is a list.
`p.adjust.method`	method for adjusting p values (see `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 `NA`s. Defaults to `getOption("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 $\mathrm{H}_{ij}: F_i(x) = F_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij}: F_i(x) \ne F_j(x), ~~ i \ne j$.

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

Scholz, F.W., Stephens, M.A. (1987) K-Sample Anderson-Darling Tests. *Journal of the American Statistical Association* **82**, 918–924.

See Also

`adKSampleTest`, `adManyOneTest`, `ad.pval`.

Examples

```
adKSampleTest(count ~ spray, InsectSprays)

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

`adKSampleTest` *Anderson-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 `NA`s. Defaults to `getOption("na.action")`.

Details

The null hypothesis, $\mathrm{H}_0 : F_1 = F_2 = \ldots = F_k$ is tested against the alternative, $\mathrm{H}_\mathrm{A} : F_i \neq F_j ~~ (i \neq j)$, with at least one unequality beeing strict.

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

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

Scholz, F.W., Stephens, M.A. (1987) K-Sample Anderson-Darling Tests. *Journal of the American Statistical Association* **82**, 918–924.

See Also

`adAllPairsTest`, `adManyOneTest`, `ad.pval`.

Examples

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

## Median Test asymptotic
medianTest(x ~ g, dat)

## Median Test with simulated p-values
set.seed(112)
medianTest(x ~ g, dat, simulate.p.value = TRUE)
```

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

Description

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

Usage

```
adManyOneTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`p.adjust.method` method for adjusting p values (see `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 `NA`s. 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. $\mathrm{H}_i: F_0 = F_i$ is tested in the two-tailed case against $\mathrm{A}_i: F_0 \ne F_i, ~~ (1 \le i \le m)$.

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.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

Scholz, F.W., Stephens, M.A. (1987) K-Sample Anderson-Darling Tests. *Journal of the American 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

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

References

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

See Also

`demo(algae)`

`barPlot` *Plotting PMCMR Objects*

Description

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

Usage

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

Arguments

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

Value

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

Note

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

Examples

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

bwsAllPairsTest *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 `NA`s. 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 $\mathrm{H}_{ij} : F_i(x) = F_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : F_i(x) \ne F_j(x), \;\; i \ne j$.

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

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

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

See Also

`bws_test`.

Examples

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

`bwsKSampleTest` *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 assymptotic 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 `NA`s. Defaults to `getOption("na.action")`.

Details

Let $X_{ij}~(1 \le i \le k,~1 \le 1 \le n_i)$ denote an identically and independently distributed variable that is obtained from an unknown continuous distribution $F_i(x)$. Let R_{ij} be the rank of X_{ij}, where X_{ij} is jointly ranked from 1 to N, $N = \sum_{i=1}^k n_i$. In the k-sample test the null hypothesis, H: $F_i = F_j$ is tested against the alternative, A: $F_i \ne F_j~~(i \ne j)$ with at least one inequality beeing strict. Murakami (2006) has generalized the two-sample Baumgartner-Weiß-Schindler test (Baumgartner et al. 1998) and proposed a modified statistic B_k^* defined by

$$B_k^* = \frac{1}{k} \sum_{i=1}^{k} \left\{ \frac{1}{n_i} \sum_{j=1}^{n_i} \frac{(R_{ij} - \mathsf{E}[R_{ij}])^2}{\mathsf{Var}[R_{ij}]} \right\},$$

where

$$\mathsf{E}[R_{ij}] = \frac{N+1}{n_i+1} j$$

and

$$\mathsf{Var}[R_{ij}] = \frac{j}{n_i+1}\left(1 - \frac{j}{n_i+1}\right)\frac{(N-n_i)(N+1)}{n_i+2}.$$

The p-values are estimated via an assymptotic boot-strap method. It should be noted that the B_k^* detects both differences in the unknown location parameters and / or differences in the unknown scale parameters of the k-samples.

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

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

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

See Also

`sample`, `bwsAllPairsTest`, `bwsManyOneTest`.

Examples

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

## Median Test asymptotic
medianTest(x ~ g, dat)

## Median Test with simulated p-values
set.seed(112)
medianTest(x ~ g, dat, simulate.p.value = TRUE)
```

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 `NA`s. 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 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. $\mathrm{H}_i: F_0 = F_i$ is tested in the two-tailed case against $\mathrm{A}_i: F_0 \ne F_i, ~~ (1 \le i \le m)$.

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: $\mathrm{H}_i: F_0 = F_i$ against $\mathrm{A}_i: F_0 \ge F_i, ~~ (1 \le i \le m)$. The modified test-statistic B* according to Neuhäuser (2001) and its corresponding Pr(>B*) or Pr(<B*) is computed by sequentally 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.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

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

Neuhäuser, M. (2001) One-Side Two-Sample and Trend Tests Based on a Modified Baumgartner-Weiss-Schindler Statistic. *J Nonparametric Stat* **13**, 729–739.

See Also

murakami_stat, murakami_cdf, bws_stat, bws_cdf.

Examples

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

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

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

## End(Not run)
```

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

Description

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

Usage

```
bwsTrendTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`nperm` number of permutations for the assymptotic 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 `NA`s. Defaults to `getOption("na.action")`.

Details

The null hypothesis, $\mathrm{H}_0 : F_1(u) = F_2(u) = \ldots = F_k(u) \;\; u \in R$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : F_1(u) \le F_2(u) \le \ldots \le F_k(u), \;\; F_1(u) < F_k(u), \;\; u \in R$.

The p-values are estimated through an assymptotic 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.

Factor labels for `g` must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

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

Neuhäuser, M. (2001) One-Side Two-Sample and Trend Tests Based on a Modified Baumgartner-Weiss-Schindler Statistic. *J Nonparametric Stat* **13**, 729–739.

See Also

`sample`, `bwsAllPairsTest`, `bwsManyOneTest`.

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
shanTest(x, g)
```

chaAllPairsNashimotoTest *All-Pairs Comparisons for Simply Ordered Mean Ranksums*

Description

Performs Nashimoto and Wright's all-pairs comparison procedure for simply ordered mean ranksums (NPT'-test and NPY'-test).

According to the authors, the procedure shall only be applied after Chacko's test (see `chackoTest`) indicates global significance.

Usage

```
chaAllPairsNashimotoTest(x, ...)

## Default S3 method:
chaAllPairsNashimotoTest(
  x,
  g,
  p.adjust.method = c(p.adjust.methods),
  alternative = c("greater", "less"),
  dist = c("Normal", "h"),
  ...
)

## S3 method for class 'formula'
chaAllPairsNashimotoTest(
  formula,
  data,
  subset,
  na.action,
  p.adjust.method = c(p.adjust.methods),
  alternative = c("greater", "less"),
  dist = c("Normal", "h"),
  ...
)
```

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. Ignored if `dist = "h"`.

`alternative` the alternative hypothesis. Defaults to `greater`.

`dist` the test distribution. Defaults to `Normal`.

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

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

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

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

Details

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

Let R_{ij} be the rank of X_{ij}, where X_{ij} is jointly ranked from $\left\{1, 2, \ldots, N\right\}, \;\; N = \sum_{i=1}^k n_i$, then the test statistics for all-pairs comparisons and a balanced design is calculated as

$$\hat{T}_{ij} = \max_{i \le m < m' \le j} \frac{\left(\bar{R}_{m'} - \bar{R}_m\right)}{\sigma_a / \sqrt{n}},$$

with $n = n_i$; $N = \sum_i^k n_i \;\; (1 \le i \le k)$, $\bar{R}_i$ the mean rank for the ith group, and the expected variance (without ties) $\sigma_a^2 = N \left(N + 1\right) / 12$.

For the NPY'-test (`dist = "h"`), if $T_{ij} > h_{k-1,\alpha,\infty}$.

For the unbalanced case with moderate imbalance the test statistic is

$$\hat{T}_{ij} = \max_{i \le m < m' \le j} \frac{\left(\bar{R}_{m'} - \bar{R}_m\right)}{\sigma_a \left(1/n_m + 1/n_{m'}\right)^{1/2}},$$

For the NPY'-test (`dist="h"`) the null hypothesis is rejected in an unbalanced design, if $\hat{T}_{ij} > h_{k,\alpha,\infty} / \sqrt{2}$. In case of a NPY'-test, the function does not return p-values. Instead the critical h-values as given in the tables of Hayter (1990) for $\alpha = 0.05$ (one-sided) are looked up according to the number of groups $(k - 1)$ and the degree of freedoms $(v = \infty)$.

For the NPT'-test (`dist = "Normal"`), the null hypothesis is rejected, if $T_{ij} > \sqrt{2} t_{\alpha,\infty} = \sqrt{2} z_\alpha$. Although Nashimoto and Wright (2005) originally did not use any p-adjustment, any method as available by `p.adjust.methods` can be selected for the adjustment of p-values estimated from the standard normal distribution.

Value

Either a list of class `"osrt"` if `dist = "h"` or a list of class `"PMCMR"` if `dist = "Normal"`.

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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

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

The function will give a warning for the unbalanced case and returns the critical value $h_{k-1,\alpha,\infty}/\sqrt{2}$ if applicable.

References

Hayter, A. J.(1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative, *J Amer Stat Assoc* **85**, 778–785.

Nashimoto, K., Wright, F.T. (2007) Nonparametric Multiple-Comparison Methods for Simply Ordered Medians. *Comput Stat Data Anal* **51**, 5068–5076.

See Also

Normal, chackoTest, NPMTest

Examples

```
## Example from Shirley (1977)
## Reaction times of mice to stimuli to their tails.
x <- 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)

## Shirley's test
## one-sided test using look-up table
shirleyWilliamsTest(x ~ g, alternative = "greater")
```

```
## Chacko's global hypothesis test for 'greater'
chackoTest(x , g)

## post-hoc test, default is standard normal distribution (NPT'-test)
summary(chaAllPairsNashimotoTest(x, g, p.adjust.method = "none"))

## same but h-distribution (NPY'-test)
chaAllPairsNashimotoTest(x, g, dist = "h")

## NPM-test
NPMTest(x, g)

## Hayter-Stone test
hayterStoneTest(x, g)

## all-pairs comparisons
hsAllPairsTest(x, g)
```

---

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

Description

Performs Chacko's test for testing against ordered alternatives.

Usage

```
chackoTest(x, ...)

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

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

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 `NA`s. Defaults to `getOption("na.action")`.

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : \theta_1 \le \theta_2 \le \ldots \le \theta_k,~\theta_1 < \theta_k$.

Let R_{ij} be the rank of X_{ij}, where X_{ij} is jointly ranked from $\left\{1, 2, \ldots, N \right\}, ~~ N = \sum_{i=1}^k n_i$, then the test statistic is calculated as

$$H = \frac{1}{\sigma_R^2} \sum_{i=1}^k n_i \left(\bar{R^*}_i - \bar{R} \right),$$

where $\bar{R^*}_i$ is the isotonic mean of the i-th group and $\sigma_R^2 = N \left(N + 1 \right) / 12$ the expected variance (without ties). H_0 is rejected, if $H > \chi_{v,\alpha}^2$ with $v = k - 1$ degree of freedom. The p-values are estimated from the chi-square distribution.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

Source

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

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

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

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

The Algorith AS 206 is available from StatLib https://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

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

The function does neither check nor correct for ties.

References

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

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)
```

```
## Shan-Young-Kang test
shanTest(x, g)
```

chenJanTest *Chen and Jan Many-to-One Comparisons Test*

Description

Performs Chen and Jan nonparametric test for contrasting increasing (decreasing) dose levels of a treatment in a randomized block design.

Usage

```
chenJanTest(y, ...)

## Default S3 method:
chenJanTest(
  y,
  groups,
  blocks,
  alternative = c("greater", "less"),
  p.adjust.method = c("single-step", "SD1", 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 `greater`.
`p.adjust.method`	method for adjusting p values (see `p.adjust`)
`...`	further arguments to be passed to or from methods.

Details

Chen's test is a non-parametric step-down trend test for testing several treatment levels with a zero control. Let there be k groups including the control and let the zero dose level be indicated with $i = 0$ and the highest dose level with $i = m$, then the following `m = k - 1` hypotheses are tested:

$$\begin{array}{ll}
\mathrm{H}_m : \theta_0 = \theta_1 = \ldots = \theta_m, & \mathrm{A}_m = \theta_0 \le \theta_1 \le \ldots \theta_m, \theta_0 < \theta_m \\
\mathrm{H}_{m-1} : \theta_0 = \theta_1 = \ldots = \theta_{m-1}, & \mathrm{A}_{m-1} = \theta_0 \le \theta_1 \le \ldots \theta_{m-1}, \theta_0 < \theta_{m-1} \\
\vdots & \vdots \\
\mathrm{H}_1 : \theta_0 = \theta_1, & \mathrm{A}_1 = \theta_0 < \theta_1
\end{array}$$

Let $Y_{ij1}, Y_{ij2}, \ldots, Y_{ijn_{ij}}$ $(i = 1, 2, \ldots, b, j = 0, 1, \ldots, k$ and $n_{ij} \ge 1)$ be a i.i.d. random variable of at least ordinal scale. Further,the zero dose control is indicated with $j = 0$.

The Mann-Whittney statistic is

$$T_{ij} = \sum_{u=0}^{j-1} \sum_{s=1}^{n_{ij}} \sum_{r=1}^{n_{iu}} I(Y_{ijs} - Y_{iur}), \qquad i = 1, 2, \ldots, b,\ j = 1, 2, \ldots, k,$$

where where the indicator function returns $I(a) = 1,\ \mathrm{if}\ a > 0, 0.5$ if$a = 0$ otherwise 0.

Let

$$N_{ij} = \sum_{s=0}^{j} n_{is} \qquad i = 1, 2, \ldots, b,\ j = 1, 2, \ldots, k,$$

and

$$T_j = \sum_{i=1}^{b} T_{ij} \qquad j = 1, 2, \ldots, k.$$

The mean and variance of T_j are

$$\mu(T_j) = \sum_{i=1}^{b} n_{ij}\ N_{ij-1}/2 \qquad \text{and}$$

$$\sigma(T_j) = \sum_{i=1}^{b} n_{ij}\ N_{ij-1} \left[(N_{ij} + 1) - \sum_{u=1}^{g_i} \left(t_u^3 - t_u \right) / \left\{ N_{ij} \left(N_{ij} - 1 \right) \right\} \right] /2,$$

with g_i the number of ties in the ith block and t_u the size of the tied group u.

The test statistic T_j^* is asymptotically multivariate normal distributed.

$$T_j^* = \frac{T_j - \mu(T_j)}{\sigma(T_j)}$$

If p.adjust.method = "single-step" than the p-values are calculated with the probability function of the multivariate normal distribution with $\Sigma = I_k$. Otherwise the standard normal distribution is used to calculate p-values and any method as available by p.adjust or by the step-down procedure as proposed by Chen (1999), if p.adjust.method = "SD1" can be used to account for α-error inflation.

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

Chen, Y.I., Jan, S.L., 2002. Nonparametric Identification of the Minimum Effective Dose for Randomized Block Designs. *Commun Stat-Simul Comput* **31**, 301–312.

See Also

Normal pmvnorm

Examples

```
## Example from Chen and Jan (2002, p. 306)
## MED is at dose level 2 (0.5 ppm SO2)
y <- c(0.2, 6.2, 0.3, 0.3, 4.9, 1.8, 3.9, 2, 0.3, 2.5, 5.4, 2.3, 12.7,
-0.2, 2.1, 6, 1.8, 3.9, 1.1, 3.8, 2.5, 1.3, -0.8, 13.1, 1.1,
12.8, 18.2, 3.4, 13.5, 4.4, 6.1, 2.8, 4, 10.6, 9, 4.2, 6.7, 35,
9, 12.9, 2, 7.1, 1.5, 10.6)
groups <- gl(4,11, labels = c("0", "0.25", "0.5", "1.0"))
blocks <- structure(rep(1:11, 4), class = "factor",
levels = c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11"))

summary(chenJanTest(y, groups, blocks, alternative = "greater"))
summary(chenJanTest(y, groups, blocks, alternative = "greater", p.adjust = "SD1"))
```

chenTest *Chen's Many-to-One Comparisons Test*

Description

Performs Chen's nonparametric test for contrasting increasing (decreasing) dose levels of a treatment.

Usage

```
chenTest(x, ...)

## Default S3 method:
chenTest(
  x,
  g,
  alternative = c("greater", "less"),
  p.adjust.method = c("SD1", p.adjust.methods),
  ...
)

## S3 method for class 'formula'
chenTest(
  formula,
  data,
  subset,
  na.action,
  alternative = c("greater", "less"),
  p.adjust.method = c("SD1", 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 `NA`s. Defaults to `getOption("na.action")`.

Details

Chen's test is a non-parametric step-down trend test for testing several treatment levels with a zero control. Let X_{0j} denote a variable with the j-th realization of the control group ($1 \leq j \leq n_0$) and X_{ij} the j-the realization in the i-th treatment group ($1 \leq i \leq k$). The variables are i.i.d. of a least ordinal scale with $F(x) = F(x_0) = F(x_i),\ (1 \leq i \leq k)$. A total of $m = k$ hypotheses can be tested:

$$\begin{array}{ll} \mathrm{H}_m : \theta_0 = \theta_1 = \ldots = \theta_m, & \mathrm{A}_m = \theta_0 \leq \theta_1 \leq \ldots \theta_m, \theta_0 < \theta_m \\ \mathrm{H}_{m-1} : \theta_0 = \theta_1 = \ldots = \theta_{m-1}, & \mathrm{A}_{m-1} = \theta_0 \leq \theta_1 \leq \ldots \theta_{m-1}, \theta_0 < \theta_{m-1} \\ \vdots & \vdots \\ \mathrm{H}_1 : \theta_0 = \theta_1, & \mathrm{A}_1 = \theta_0 < \theta_1 \end{array}$$

The statistics T_i are based on a Wilcoxon-type ranking:

$$T_i = \sum_{j=0}^{i=1} \sum_{u=1}^{n_i} \sum_{v=1}^{n_j} I(x_{iu} - x_{jv}), \qquad (1 \leq i \leq k),$$

where the indicator function returns $I(a) = 1,\ \text{if}\ a > 0, 0.5\ \text{if} a = 0$ otherwise 0.

The expected ith mean is

$$\mu(T_i) = n_i N_{i-1}/2,$$

with $N_j = \sum_{j=0}^{i} n_j$ and the ith variance:

$$\sigma^2(T_i) = n_i N_{i-1}/12 \left\{ N_i + 1 - \sum_{j=1}^{g} t_j \left(t_j^2 - 1\right) / \left[N_i \left(N_i - 1\right)\right] \right\}.$$

The test statistic T_i^* is asymptotically standard normal

$$T_i^* = \frac{T_i - \mu(T_i)}{\sqrt{\sigma^2(T_i)}}, \qquad (1 \leq i \leq k).$$

The p-values are calculated from the standard normal distribution. The p-values can be adjusted with any method as available by `p.adjust` or by the step-down procedure as proposed by Chen (1999), if `p.adjust.method = "SD1"`.

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

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see order, factor).

References

Chen, Y.-I., 1999, Nonparametric Identification of the Minimum Effective Dose. *Biometrics* **55**, 1236–1240. doi:10.1111/j.0006341X.1999.01236.x

See Also

wilcox.test, Normal

Examples

```
## Chen, 1999, p. 1237,
## Minimum effective dose (MED)
## is at 2nd dose level
df <- data.frame(x = c(23, 22, 14,
27, 23, 21,
28, 37, 35,
41, 37, 43,
28, 21, 30,
16, 19, 13),
g = gl(6, 3))
levels(df$g) <- 0:5
ans <- chenTest(x ~ g, data = df, alternative = "greater",
                p.adjust.method = "SD1")
summary(ans)
```

Cochran *Cochran's distribution*

Description

Distribution function and quantile function for Cochran's distribution.

Usage

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

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

Arguments

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

Value

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

References

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

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.*. doi:10.1007/s101820110185y.

See Also

`FDist`

Examples

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

`cochranTest` *Cochran Test*

Description

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

Usage

```
cochranTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`alternative` the alternative hypothesis. Defaults to `"greater"`

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

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

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

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

Details

For normally distributed data the null hypothesis, $\mathrm{H}_0 : \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2$ is tested against the alternative (greater) $\mathrm{H}_\mathrm{A} : \sigma_p > \sigma_i \;\; (i \leq k, i \neq p)$ with at least one inequality being strict.

The p-value is computed with the function `pcochran`.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

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

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.*. doi:10.1007/s101820110185y.

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 `NA`s. Defaults to `getOption("na.action")`.

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : \theta_1 \le \theta_2 \le \ldots \le \theta_k,\ \theta_1 < \theta_k$.

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.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
shanTest(x, g)
```

Dgrubbs *Grubbs D* distribution*

Description

Distribution function for Grubbs D* distribution.

Usage

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

Arguments

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

Value

pgrubbs gives the distribution function

References

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

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic, *Adv. Stat. Anal.*. doi:10.1007/s101820110185y.

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 $(1 \le i \le k)$. Further, let the increasingly ordered realizations denote $x_{(1)} \le x_{(2)} \le \ldots \le x_{(n)}$. 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, $\mathrm{H}_0 : \Delta = 0$ is tested against the alternative, $\mathrm{H}_\mathrm{A} : \Delta > 0$.

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, $\mathrm{H}_\mathrm{A} : \Delta < 0$.

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

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

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

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.*. doi:10.1007/s101820110185y.

Examples

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

dscfAllPairsTest *Multiple Comparisons of Mean Rank Sums*

Description

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

Usage

```
dscfAllPairsTest(x, ...)

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

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

Arguments

`x`	a numeric vector of data values, or a list of numeric data vectors.
`...`	further arguments to be passed to or from methods.
`g`	a vector or factor object giving the group for the corresponding elements of `"x"`. Ignored with a warning if `"x"` is a list.
`formula`	a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
`data`	an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
`subset`	an optional vector specifying a subset of observations to be used.
`na.action`	a function which indicates what should happen when the data contain `NA`s. Defaults to `getOption("na.action")`.

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 $\mathrm{H}_{ij} : F_i(x) = F_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : F_i(x) \ne F_j(x), \;\; i \ne j$. 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 distriburtion.

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

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

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

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

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 `NA`s. Defaults to `getOption("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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j ~~ (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Duncan's all-pairs test statistics are given by

$$t_{(i)(j)} \frac{\bar{X}_{(i)} - \bar{X}_{(j)}}{s_{\mathrm{in}} \left(r\right)^{1/2}}, ~~ (i < j)$$

with s^2_{in} the within-group ANOVA variance, $r = k / \sum_{i=1}^k n_i$ and $\bar{X}_{(i)}$ the increasingly ordered means $1 \le i \le k$. The null hypothesis is rejected if

$$\Pr \left\{ |t_{(i)(j)}| \ge q_{vm'\alpha'} | \mathrm{H} \right\}_{(i)(j)} = \alpha' = \min \left\{ 1,~ 1 - (1 - \alpha)^{(1/(m'-1))} \right\},$$

with $v = N - k$ degree of freedom, the range $m' = 1 + |i - j|$ and α' the Bonferroni adjusted alpha-error. 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

Duncan, D. B. (1955) Multiple range and multiple F tests, *Biometrics* **11**, 1–42.

See Also

Tukey, TukeyHSD tukeyTest

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 <- 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 `NA`s. 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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j ~~ (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Dunnett T3 all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{\left(s^2_j / n_j + s^2_i / n_i\right)^{1/2}}, ~~ (i \ne j)$$

with s^2_i the variance of the i-th group. The null hypothesis is rejected (two-tailed) if

$$\Pr \left\{ |t_{ij}| \ge T_{v_{ij} \rho_{ij} \alpha'/2} | \mathrm{H} \right\}_{ij} = \alpha,$$

with Welch's approximate solution for calculating the degree of freedom.

$$v_{ij} = \frac{\left(s^2_i / n_i + s^2_j / n_j\right)^2}{s^4_i / n^2_i \left(n_i - 1\right) + s^4_j / n^2_j \left(n_j - 1\right)}.$$

The p-values are computed from the studentized maximum modulus distribution that is the equivalent of the multivariate t distribution with $\rho_{ii} = 1, ~ \rho_{ij} = 0 ~ (i \ne j)$. The function `pmvt` is used to calculate the p-values.

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

See Also

pmvt

Examples

```
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,
  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.

`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 `NA`s. 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. Let X_{0j} denote a continuous random variable with the j-the realization of the control group ($1 \le j \le n_0$) and X_{ij} the j-the realization in the i-th treatment group ($1 \le i \le k$). Furthermore, the total sample size is $N = n_0 + \sum_{i=1}^k n_i$. A total of $m = k$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_i : \mu_i = \mu_0$ is tested against the alternative $\mathrm{A}_i : \mu_i \ne \mu_0$ (two-tailed). Dunnett's test statistics are given by

$$t_i \frac{\bar{X}_i - \bar{X}_0}{s_{\mathrm{in}} \left(1/n_0 + 1/n_i\right)^{1/2}}, \quad (1 \le i \le k)$$

with s^2_{in} the within-group ANOVA variance. The null hypothesis is rejected if $|t_{ij}| > |T_{kv\rho\alpha}|$ (two-tailed), with $v = N - k$ degree of freedom and rho the correlation:

$$\rho_{ij} = \sqrt{\frac{n_i n_j}{\left(n_i + n_0\right)\left(n_j + n_0\right)}} \quad (i \ne j).$$

The p-values are computed with the function `pDunnett` that is a wrapper to the 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

Dunnett, C. W. (1955) A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association* **50**, 1096–1121.

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

See Also

`pmvt` `pDunnett`

Examples

```
fit <- aov(Y ~ DOSE, data = trout)
shapiro.test(residuals(fit))
bartlett.test(Y ~ DOSE, data = trout)

## works with fitted object of class aov
summary(dunnettTest(fit, alternative = "less"))
```

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

Description

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

Usage

```
durbinAllPairsTest(y, ...)

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

Arguments

`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 $\mathrm{H}_{ij} : \theta_i = \theta_j$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \theta_i \ne \theta_j, \;\; i \ne j$.

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

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

See Also

`durbinTest`

Examples

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

`durbinTest` *Durbin Test*

Description

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

Usage

```
durbinTest(y, ...)

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

Arguments

`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, $\mathrm{H}_0 : \theta_i = \theta_j ~ (1 \le i < j \le k)$, is tested against the alternative that at least one $\theta_i \ne \theta_j$.

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

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

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

Examples

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

flignerWolfeTest *Testing Several Treatments With One Control*

Description

Performs Fligner-Wolfe non-parametric test for simultaneous testing of several locations of treatment groups against the location of the control group.

Usage

```
flignerWolfeTest(x, ...)

## Default S3 method:
flignerWolfeTest(
  x,
  g,
  alternative = c("greater", "less"),
  dist = c("Wilcoxon", "Normal"),
  ...
)

## S3 method for class 'formula'
flignerWolfeTest(
  formula,
  data,
  subset,
  na.action,
  alternative = c("greater", "less"),
  dist = c("Wilcoxon", "Normal"),
  ...
)
```

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"`.

`dist` the test distribution. Defaults to `"Wilcoxon"`.

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

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

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

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

Details

For a one-factorial layout with non-normally distributed residuals the Fligner-Wolfe test can be used.

Let there be $k-1$-treatment groups and one control group, then the null hypothesis, $\mathrm{H}_0: \theta_i - \theta_c = 0 ~ (1 \le i \le k-1)$ is tested against the alternative (greater), $\mathrm{A}_1: \theta_i - \theta_c > 0 ~ (1 \le i \le k-1)$, with at least one inequality being strict.

Let n_c denote the sample size of the control group, $N^t = \sum_{i=1}^{k-1} n_i$ the sum of all treatment sample sizes and $N = N^t + n_c$. The test statistic without taken ties into account is

$$W = \sum_{j=1}^{k-1} \sum_{i=1}^{n_i} r_{ij} - \frac{N^t \left(N^t + 1\right)}{2}$$

with r_{ij} the rank of variable x_{ij}. The null hypothesis is rejected, if $W > W_{\alpha,m,n}$ with $m = N^t$ and $n = n_c$.

In the presence of ties, the statistic is

$$\hat{z} = \frac{W - n_c N^t / 2}{s_W},$$

where

$$s_W = \frac{n_c N^t}{12N\left(N-1\right)} \sum_{j=1}^{g} t_j \left(t_j^2 - 1\right),$$

with g the number of tied groups and t_j the number of tied values in the jth group. The null hypothesis is rejected, if $\hat{z} > z_\alpha$ (as cited in EPA 2006).

If `dist = Wilcoxon`, then the p-values are estimated from the `Wilcoxon` distribution, else the `Normal` distribution is used. The latter can be used, if ties are present.

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

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

EPA (2006) *Data Quality Assessment: Statistical Methods for Practitioners* (Guideline No. EPA QA/G-9S), US-EPA.

Fligner, M.A., Wolfe, D.A. (1982) Distribution-free tests for comparing several treatments with a control. *Stat Neerl* **36**, 119–127.

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
shanTest(x, g)
```

---

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

Description

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

Usage

```
frdAllPairsConoverTest(y, ...)

## Default S3 method:
```

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

Arguments

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

Details

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

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

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

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

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

See Also

friedmanTest, friedman.test, frdAllPairsExactTest, frdAllPairsMillerTest, frdAllPairsNemenyiTest, frdAllPairsSiegelTest

Examples

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

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

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

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

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

## 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)
```

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, $\mathrm{H}_{ij} : \theta_i = \theta_j$, is tested in the two-tailed case against the alternative, $\mathrm{A}_{ij} : \theta_i \ne \theta_j, \;\; i \ne j$.

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

Source

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

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

References

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

See Also

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

Examples

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

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

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

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

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

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

Description

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

Usage

```
frdAllPairsMillerTest(y, ...)

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

Arguments

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, $\mathrm{H}_{ij} : \theta_i = \theta_j$, is tested in the two-tailed case against the alternative, $\mathrm{A}_{ij} : \theta_i \ne \theta_j, \;\; i \ne j$.

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

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

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

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

See Also

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

Examples

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

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

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

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

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

## 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)
```

`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 `NA`s. 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 $m = k(k-1)/2$ hypotheses can be tested. The null hypothesis, $\mathrm{H}_{ij} : \theta_i = \theta_j$, is tested in the two-tailed case against the alternative, $\mathrm{A}_{ij} : \theta_i \ne \theta_j, \;\; i \ne j$.

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

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

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

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

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 $m = k(k-1)/2$ hypotheses can be tested. The null hypothesis, $\mathrm{H}_{ij} : \theta_i = \theta_j$, is tested in the two-tailed case against the alternative, $\mathrm{A}_{ij} : \theta_i \ne \theta_j, \;\; i \ne j$.

The p-values are computed from the standard normal distribution. Any method as implemented in `p.adjust` can be used for p-value adjustment.

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

See Also

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

Examples

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

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

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

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

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

## 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)
```

frdHouseTest *House Test*

Description

Performs House nonparametric equivalent of William’s test for contrasting increasing dose levels of a treatment in a complete randomized block design.

Usage

```
frdHouseTest(y, ...)

## Default S3 method:
frdHouseTest(y, groups, blocks, alternative = c("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 `greater`.

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

Details

House test is a non-parametric step-down trend test for testing several treatment levels with a zero control. Let there be k groups including the control and let the zero dose level be indicated with $i = 0$ and the highest dose level with $i = m$, then the following `m = k - 1` hypotheses are tested:

$$\begin{array}{ll}
\mathrm{H}_{m}: \theta_0 = \theta_1 = \ldots = \theta_m, & \mathrm{A}_{m} = \theta_0 \le \theta_1 \le \ldots \theta_m, \theta_0 < \theta_m \\
\mathrm{H}_{m-1}: \theta_0 = \theta_1 = \ldots = \theta_{m-1}, & \mathrm{A}_{m-1} = \theta_0 \le \theta_1 \le \ldots \theta_{m-1}, \theta_0 < \theta_{m-1} \\
\vdots & \vdots \\
\mathrm{H}_{1}: \theta_0 = \theta_1, & \mathrm{A}_{1} = \theta_0 < \theta_1
\end{array}$$

Let Y_{ij} $(1 \le i \le n, 0 \le j \le k)$ be a i.i.d. random variable of at least ordinal scale. Further, let $\bar{R}_0,\ \bar{R}_1, \ldots,\ \bar{R}_k$ be Friedman's average ranks and set $\bar{R}_0^*, \le \ldots \le \bar{R}_k^*$ to be its isotonic regression estimators under the order restriction $\theta_0 \le \ldots \le \theta_k$.

The statistics is

$$T_j = \left(\bar{R}_j^* - \bar{R}_0\right)\ \left[\left(V_j - H_j\right)(2/n)\right]^{-1/2} \qquad (1 \le j \le k),$$

with

$$V_j = (j + 1)\ (j + 2)\,/12$$

and

$$H_j = \left(t^3 - t\right) / \left(12 j n\right),$$

where t is the number of tied ranks.

The critical $t'_{i,v,\alpha}$-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 (j).

For the comparison of the first dose level ($j = 1$) with the control, the critical z-value from the standard normal distribution is used (`Normal`).

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

Chen, Y.-I., 1999. Rank-Based Tests for Dose Finding in Nonmonotonic Dose–Response Settings. *Biometrics* **55**, 1258–1262. doi:10.1111/j.0006341X.1999.01258.x

House, D.E., 1986. A Nonparametric Version of Williams' Test for Randomized Block Design. *Biometrics* **42**, 187–190.

See Also

`friedmanTest`, `friedman.test`, `frdManyOneExactTest`, `frdManyOneDemsarTest`

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
summary(frdManyOneDemsarTest(y=y, p.adjust = "bonferroni",
                     alternative = "greater"))

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

## Nemenyi's many-one test
summary(frdManyOneNemenyiTest(y=y, alternative = "greater"))

## House test
frdHouseTest(y, alternative = "greater")
```

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

Description

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

Usage

```
frdManyOneDemsarTest(y, ...)

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

Arguments

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

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

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

`alternative` the alternative hypothesis. Defaults to `two.sided`.

`p.adjust.method`
method for adjusting p values (see `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. $\mathrm{H}_i : \theta_0 = \theta_i$ is tested in the two-tailed case against $\mathrm{A}_i : \theta_0 \ne \theta_i, \;\; (1 \le i \le m)$.

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

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

See Also

friedmanTest, friedman.test, frdManyOneExactTest, frdManyOneNemenyiTest.

Examples

```
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics
## (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
## Assume A is the control.

y <- matrix(c(
3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
26.65),nrow=6, ncol=6,
dimnames=list(1:6, LETTERS[1:6]))

## Global Friedman test
friedmanTest(y)

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

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

```
## Nemenyi's many-one test
summary(frdManyOneNemenyiTest(y=y, alternative = "greater"))

## House test
frdHouseTest(y, alternative = "greater")
```

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 unreplicated complete block design with non-normally distributed residuals, an exact test can be performed on Friedman-type ranked data.

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

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

See Also

friedmanTest, friedman.test, frdManyOneDemsarTest, frdManyOneNemenyiTest.

Examples

```
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics
## (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
## Assume A is the control.

y <- matrix(c(
3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
26.65),nrow=6, ncol=6,
dimnames=list(1:6, LETTERS[1:6]))

## Global Friedman test
friedmanTest(y)

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

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

```
## Nemenyi's many-one test
summary(frdManyOneNemenyiTest(y=y, alternative = "greater"))

## House test
frdHouseTest(y, alternative = "greater")
```

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

Description

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

Usage

```
frdManyOneNemenyiTest(y, ...)

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

Arguments

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 unreplicated complete block design with non-normally distributed residuals, Nemenyi's test can be performed on Friedman-type ranked data.

Let there be k groups including the control, then the number of treatment levels is $m = k - 1$. A total of m pairwise comparisons can be performed between the i-th treatment level and the control. $\mathrm{H}_i : \theta_0 = \theta_i$ is tested in the two-tailed case against $\mathrm{A}_i : \theta_0 \ne \theta_i, \;\; (1 \le i \le m)$.

The p-values are computed from the multivariate normal distribution. As pmvnorm applies a numerical method, the estimated p-values are seet 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

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

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

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

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

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

See Also

friedmanTest, friedman.test, frdManyOneExactTest, frdManyOneDemsarTest pmvnorm, 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
```

```
summary(frdManyOneDemsarTest(y=y, p.adjust = "bonferroni",
                     alternative = "greater"))

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

## Nemenyi's many-one test
summary(frdManyOneNemenyiTest(y=y, alternative = "greater"))

## House test
frdHouseTest(y, alternative = "greater")
```

`friedmanTest` *Friedman Rank Sum Test*

Description

Performs a Friedman rank sum test. The null hypothesis $\mathrm{H}_0: \theta_i = \theta_j \;\; (i \neq j)$ is tested against the alternative $\mathrm{H}_{\mathrm{A}}: \theta_i \neq \theta_j$, with at least one inequality beeing 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 anf Iman (1981). Friedman's test statistic is assymptotically 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

Conover, W.J., Iman, R.L. (1981) Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics. *Am Stat* **35**, 124–129.

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

See Also

friedman.test

Examples

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

## Chisquare distribution
friedmanTest(RoundingTimes)

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

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

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

gamesHowellTest *Games-Howell Test*

Description

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

Usage

```
gamesHowellTest(x, ...)

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

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

## 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 `NA`s. Defaults to `getOption("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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j \;\; (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Games-Howell Test all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{\left(s_j^2/n_j + s_i^2/n_i\right)^{1/2}}, \;\; (i \ne j)$$

with s_i^2 the variance of the i-th group. The null hypothesis is rejected (two-tailed) if

$$\mathrm{Pr} \left\{ |t_{ij}| \sqrt{2} \ge q_{m v_{ij} \alpha} | \mathrm{H} \right\}_{ij} = \alpha,$$

with Welch's approximate solution for calculating the degree of freedom.

$$v_{ij} = \frac{\left(s_i^2/n_i + s_j^2/n_j\right)^2}{s_i^4/n_i^2 \left(n_i - 1\right) + s_j^4/n_j^2 \left(n_j - 1\right)}.$$

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.

See Also

Tukey

Examples

```
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 <- gamesHowellTest(fit)
summary(res)
summaryGroup(res)
```

---

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.

References

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

Examples

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

out <- gesdTest(x, 10)

## print method
out

## summary method
summary(out)
```

goreTest *Gore Test*

Description

Performs Gore's test. The null hypothesis $\mathrm{H}_0: \theta_i = \theta_j \;\; (i \neq j)$ is tested against the alternative $\mathrm{H}_\mathrm{A}: \theta_i \neq \theta_j$, with at least one inequality beeing 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 ore more observations per cell. The statistic is assymptotically chi-squared distributed.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

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

See Also

friedmanTest, skillingsMackTest, durbinTest

Examples

```
## Crop Yield of 3 varieties on two
## soil classes
X <-c("130,A,Light
115,A,Light
123,A,Light
142,A,Light
117,A,Heavy
125,A,Heavy
139,A,Heavy
108,B,Light
114,B,Light
124,B,Light
106,B,Light
91,B,Heavy
111,B,Heavy
110,B,Heavy
155,C,Light
146,C,Light
151,C,Light
165,C,Light
97,C,Heavy
108,C,Heavy")
con <- textConnection(X)
x <- read.table(con, header=FALSE, sep=",")
close(con)
colnames(x) <- c("Yield", "Variety", "SoilType")
goreTest(y = x$Yield, groups = x$Variety, blocks = x$SoilType)
```

Grubbs *Grubbs distribution*

Description

Distribution function and quantile function for Grubbs distribution.

Usage

```
qgrubbs(p, n)

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

Arguments

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

Value

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

References

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

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.*. doi:10.1007/s101820110185y.

See Also

`TDist`

Examples

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

grubbsTest *Grubbs Outlier Test*

Description

Performs Grubbs single outlier test.

Usage

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

Arguments

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

Details

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

$$x_{(i)} = \left\{ \begin{array}{ll} \mu + \epsilon_{(i)}, & i = 1, \ldots, n - 1 \\ \mu + \Delta + \epsilon_{(n)} & \end{array} \right.$$

with $\epsilon \approx N(0, \sigma)$. The null hypothesis, $\mathrm{H}_0 : \Delta = 0$ is tested against the alternative, $\mathrm{H}_{\mathrm{A}} : \Delta > 0$.

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

$$x_{(i)} = \left\{ \begin{array}{ll} \mu + \Delta + \epsilon_{(1)} & \\ \mu + \epsilon_{(i)}, & i = 2, \ldots, n \end{array} \right.$$

The null hypothesis is tested against the alternative, $\mathrm{H}_{\mathrm{A}} : \Delta < 0$.

The p-value is computed with the function `pgrubbs`.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

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

Wilrich, P.-T. (2011) Critical values of Mandel's h and k, Grubbs and the Cochran test statistic. *Adv. Stat. Anal.*. doi:10.1007/s101820110185y.

Examples

```
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

```
GSTTest(x, ...)

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

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

Details

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

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

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

H.F.L. Meyer-Bahlburg (1970), A nonparametric test for relative spread in k unpaired samples, *Metrika* **15**, 23–29.

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)
```

hartleyTest *Hartley's Maximum F-Ratio Test of Homogeneity of Variances*

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 `NA`s. Defaults to `getOption("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 hartleyTest(x) to perform the test. If the samples are not yet contained in a list, use hartleyTest(list(x, ...)).

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 pmaxFratio 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

Hartley, H.O. (1950) The maximum F-ratio as a short cut test for heterogeneity of variance, *Biometrika* **37**, 308–312.

See Also

bartlett.test, pmaxFratio

Examples

```
hartleyTest(count ~ spray, data = InsectSprays)
```

hayterStoneTest *Hayter-Stone Test*

Description

Performs the non-parametric Hayter-Stone procedure to test against an monotonically increasing alternative.

Usage

```
hayterStoneTest(x, ...)

## Default S3 method:
hayterStoneTest(
  x,
  g,
  alternative = c("greater", "less"),
  method = c("look-up", "boot", "asympt"),
  nperm = 10000,
  ...
)

## S3 method for class 'formula'
hayterStoneTest(
  formula,
  data,
  subset,
  na.action,
  alternative = c("greater", "less"),
  method = c("look-up", "boot", "asympt"),
  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 `greater`.
`method`	a character string specifying the test statistic to use. Defaults to `"look-up"` that uses published Table values.
`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 `NA`s. Defaults to `getOption("na.action")`.

Details

Let X be an identically and idepentendly distributed variable that was n times observed at k increasing treatment levels. Hayter and Stone (1991) proposed a non-parametric procedure to test the null hypothesis, H: $\theta_i = \theta_j \ \ (i < j \leq k)$ against a simple order alternative, A: $\theta_i < \theta_j$, with at least one inequality being strict.

The statistic for a global test is calculated as,

$$h = \max_{1 \leq i < j \leq k} \frac{2 \sqrt{6} \left(U_{ij} - n_i n_j / 2\right)}{\sqrt{n_i n_j \left(n_i + n_j + 1\right)}},$$

with the Mann-Whittney counts:

$$U_{ij} = \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} I \left\{ x_{ia} < x_{ja} \right\}.$$

Under the large sample approximation, the test statistic h is distributed as $h_{k,\alpha,v}$. Thus, the null hypothesis is rejected, if $h > h_{k,\alpha,v}$, with $v = \infty$ degree of freedom.

If `method = "look-up"` the function will not return p-values. Instead the critical h-values as given in the tables of Hayter (1990) for $\alpha = 0.05$ (one-sided) are looked up according to the number of groups (k) and the degree of freedoms ($v = \infty$).

If `method = "boot"` an asymptotic permutation test is conducted and a p-value is returned.

If `method = "asympt"` is selected the asymptotic p-value is estimated as implemented in the function `pHayStonLSA` of the package **NSM3**.

Value

Either a list of class `htest` or a list with class `"osrt"` that contains 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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

Source

If method = "asympt" is selected, this function calls an internal probability function pHS. The GPL-2 code for this function was taken from pHayStonLSA of the the package **NSM3**:

Grant Schneider, Eric Chicken and Rachel Becvarik (2020) NSM3: Functions and Datasets to Accompany Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods, Third Edition. R package version 1.15. https://CRAN.R-project.org/package=NSM3

References

Hayter, A. J.(1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative, *J Amer Stat Assoc* **85**, 778–785.

Hayter, A.J., Stone, G. (1991) Distribution free multiple comparisons for monotonically ordered treatment effects. *Austral J Statist* **33**, 335–346.

See Also

osrtTest, hsAllPairsTest, sample, pHayStonLSA

Examples

```
## Example from Shirley (1977)
## Reaction times of mice to stimuli to their tails.
x <- 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)

## Shirley's test
## one-sided test using look-up table
shirleyWilliamsTest(x ~ g, alternative = "greater")

## Chacko's global hypothesis test for 'greater'
chackoTest(x , g)

## post-hoc test, default is standard normal distribution (NPT'-test)
summary(chaAllPairsNashimotoTest(x, g, p.adjust.method = "none"))

## same but h-distribution (NPY'-test)
chaAllPairsNashimotoTest(x, g, dist = "h")

## NPM-test
NPMTest(x, g)

## Hayter-Stone test
hayterStoneTest(x, g)

## all-pairs comparisons
hsAllPairsTest(x, g)
```

hsAllPairsTest *Hayter-Stone All-Pairs Comparison Test*

Description

Performs the non-parametric Hayter-Stone all-pairs procedure to test against monotonically increasing alternatives.

Usage

```
hsAllPairsTest(x, ...)

## Default S3 method:
hsAllPairsTest(
  x,
  g,
  alternative = c("greater", "less"),
  method = c("look-up", "boot", "asympt"),
  nperm = 10000,
  ...
)

## S3 method for class 'formula'
hsAllPairsTest(
  formula,
  data,
  subset,
  na.action,
  alternative = c("greater", "less"),
  method = c("look-up", "boot", "asympt"),
  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 `greater`.
`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 `NA`s. Defaults to `getOption("na.action")`.

Details

Let X be an identically and idepentendly distributed variable that was n times observed at k increasing treatment levels. Hayter and Stone (1991) proposed a non-parametric procedure to test the null hypothesis, H: $\theta_i = \theta_j ~~ (i < j \leq k)$ against a simple order alternative, A: $\theta_i < \theta_j$.

The statistic for all-pairs comparisons is calculated as,

$$S_{ij} = \frac{2 \sqrt{6} \left(U_{ij} - n_i n_j / 2\right)}{\sqrt{n_i n_j \left(n_i + n_j + 1\right)}},$$

with the Mann-Whittney counts:

$$U_{ij} = \sum_{a=1}^{n_i} \sum_{b=1}^{n_j} I \left\{x_{ia} < x_{ja}\right\}.$$

Under the large sample approximation, the test statistic S_{ij} is distributed as $h_{k,\alpha,v}$. Thus, the null hypothesis is rejected, if $S_{ij} > h_{k,\alpha,v}$, with $v = \infty$ degree of freedom.

If `method = "look-up"` the function will not return p-values. Instead the critical h-values as given in the tables of Hayter (1990) for $\alpha = 0.05$ (one-sided) are looked up according to the number of groups (k) and the degree of freedoms ($v = \infty$).

If `method = "boot"` an asymetric permutation test is conducted and p-values are returned.

If `method = "asympt"` is selected the asymptotic p-value is estimated as implemented in the function `pHayStonLSA` of the package **NSM3**.

Value

Either a list of class `"PMCMR"` or a list with class `"osrt"` that contains 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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

Source

If method = "asympt" is selected, this function calls an internal probability function pHS. The GPL-2 code for this function was taken from pHayStonLSA of the the package **NSM3**:

Grant Schneider, Eric Chicken and Rachel Becvarik (2020) NSM3: Functions and Datasets to Accompany Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods, Third Edition. R package version 1.15. https://CRAN.R-project.org/package=NSM3

References

Hayter, A. J.(1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative, *Journal of the American Statistical Association* **85**, 778–785.

Hayter, A.J., Stone, G. (1991) Distribution free multiple comparisons for monotonically ordered treatment effects. *Austral J Statist* **33**, 335–346.

See Also

hayterStoneTest sample

Examples

```
## Example from Shirley (1977)
## Reaction times of mice to stimuli to their tails.
x <- 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)

## Shirley's test
## one-sided test using look-up table
shirleyWilliamsTest(x ~ g, alternative = "greater")

## Chacko's global hypothesis test for 'greater'
chackoTest(x , g)
```

```
## post-hoc test, default is standard normal distribution (NPT'-test)
summary(chaAllPairsNashimotoTest(x, g, p.adjust.method = "none"))

## same but h-distribution (NPY'-test)
chaAllPairsNashimotoTest(x, g, dist = "h")

## NPM-test
NPMTest(x, g)

## Hayter-Stone test
hayterStoneTest(x, g)

## all-pairs comparisons
hsAllPairsTest(x, g)
```

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

Description

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

Usage

```
johnsonTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`alternative` the alternative hypothesis. Defaults to `"two.sided"`.

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

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

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

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

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : \theta_1 \le \theta_2 \le \ldots \le \theta_k,\ \theta_1 < \theta_k$.

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.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

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

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
shanTest(x, g)
```

---

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

Description

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

Usage

```
jonckheereTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`alternative` the alternative hypothesis. Defaults to `"two.sided"`.

`continuity` logical indicator whether a continuity correction shall be performed. Defaults to `FALSE`.

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

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

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

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

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : \theta_1 \le \theta_2 \le \ldots \le \theta_k,\ \theta_1 < \theta_k$.

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:

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

Note

`jonckheereTest(x, g, alternative = "two.sided", continuity = TRUE)` is equivalent to

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

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

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

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
shanTest(x, g)
```

kruskalTest *Kruskal-Wallis Rank Sum Test*

Description

Performs a Kruskal-Wallis rank sum test.

Usage

```
kruskalTest(x, ...)

## Default S3 method:
kruskalTest(x, g, dist = c("Chisquare", "KruskalWallis", "FDist"), ...)

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

Arguments

`x`	a numeric vector of data values, or a list of numeric data vectors.
`...`	further arguments to be passed to or from methods.
`g`	a vector or factor object giving the group for the corresponding elements of `"x"`. Ignored with a warning if `"x"` is a list.
`dist`	the test distribution. Defaults's to `"Chisquare"`.
`formula`	a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
`data`	an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
`subset`	an optional vector specifying a subset of observations to be used.
`na.action`	a function which indicates what should happen when the data contain `NA`s. Defaults to `getOption("na.action")`.

Details

For one-factorial designs with non-normally distributed residuals the Kruskal-Wallis rank sum test can be performed to test the $\mathrm{H}_0 : F_1(x) = F_2(x) = \ldots = F_k(x)$ against the $\mathrm{H}_\mathrm{A} : F_i(x) \neq F_j(x)$ $(i \neq j)$ with at least one strict inequality.

Let R_{ij} be the joint rank of X_{ij}, with $R_{(1)(1)} = 1, \ldots, R_{(n)(n)} = N, \quad N = \sum_{i=1}^k n_i$, The test statistic is calculated as

$$H = \sum_{i=1}^k n_i \left(\bar{R}_i - \bar{R}\right) / \sigma_R,$$

with the mean rank of the i-th group

$$\bar{R}_i = \sum_{j=1}^{n_i} R_{ij} / n_i,$$

the expected value

$$\bar{R} = \left(N + 1\right) / 2$$

and the expected variance as

$$\sigma_R^2 = N \left(N + 1\right) / 12.$$

In case of ties the statistic H is divided by $\left(1 - \sum_{i=1}^r t_i^3 - t_i\right) / \left(N^3 - N\right)$

According to Conover and Imam (1981), the statistic H is related to the F-quantile as

$$F = \frac{H / \left(k - 1\right)}{\left(N - 1 - H\right) / \left(N - k\right)}$$

which is equivalent to a one-way ANOVA F-test using rank transformed data (see examples).

The function provides three different `dist` for p-value estimation:

Chisquare p-values are computed from the Chisquare distribution with $v = k - 1$ degree of freedom.

KruskalWallis p-values are computed from the pKruskalWallis of the package **SuppDists**.

FDist p-values are computed from the FDist distribution with $v_1 = k - 1, \; v_2 = N - k$ degree 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.

References

Conover, W.J., Iman, R.L. (1981) Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics. *Am Stat* **35**, 124–129.

Kruskal, W.H., Wallis, W.A. (1952) Use of Ranks in One-Criterion Variance Analysis. *J Am Stat Assoc* **47**, 583–621.

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

See Also

kruskal.test, pKruskalWallis, Chisquare, FDist

Examples

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

## Median Test asymptotic
medianTest(x ~ g, dat)

## Median Test with simulated p-values
set.seed(112)
medianTest(x ~ g, dat, simulate.p.value = TRUE)
```

---

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`).

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

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

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

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

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 $\mathrm{H}_{ij} : \mu_i(x) = \mu_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \mu_i(x) \neq \mu_j(x), \;\; i \neq j$.

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

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

See Also

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

Examples

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

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

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

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

## Brown-Mood all-pairs median test
ans <- medianAllPairsTest(count ~ spray, data = InsectSprays)
summary(ans)
```

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

Description

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

Usage

```
kwAllPairsDunnTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`p.adjust.method` method for adjusting p values (see `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 `NA`s. 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 $\mathrm{H}_{ij} : \mu_i(x) = \mu_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \mu_i(x) \neq \mu_j(x), \;\; i \neq j$.

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

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

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

See Also

Normal, p.adjust, kruskalTest, kwAllPairsConoverTest, kwAllPairsNemenyiTest

Examples

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

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

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

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

## Brown-Mood all-pairs median test
ans <- medianAllPairsTest(count ~ spray, data = InsectSprays)
summary(ans)
```

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

Description

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

Usage

```
kwAllPairsNemenyiTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

`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 `NA`s. 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 $m = k(k-1)/2$ hypotheses can be tested. The null hypothesis $\mathrm{H}_{ij} : \theta_i(x) = \theta_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \theta_i(x) \neq \theta_j(x), \;\; i \neq j$.

Let R_{ij} be the rank of X_{ij}, where X_{ij} is jointly ranked from $\left\{1, 2, \ldots, N\right\}, \;\; N = \sum_{i=1}^k n_i$, then the test statistic under the absence of ties is calculated as

$$t_{ij} = \frac{\bar{R}_j - \bar{R}_i}{\sigma_R \left(1/n_i + 1/n_j\right)^{1/2}} \qquad \left(i \neq j\right),$$

with $\bar{R}_j, \bar{R}_i$ the mean rank of the i-th and j-th group and the expected variance as

$$\sigma_R^2 = N \left(N + 1\right)/12.$$

A pairwise difference is significant, if $|t_{ij}|/\sqrt{2} > q_{kv}$, with k the number of groups and $v = \infty$ the degree of freedom.

Sachs(1997) has given a modified approach for Nemenyi's test in the presence of ties for $N > 6, k > 4$ provided that the `kruskalTest` indicates significance: In the presence of ties, the test statistic is corrected according to $\hat{t}_{ij} = t_{ij}/C$, with

$$C = 1 - \frac{\sum_{i=1}^r t_i^3 - t_i}{N^3 - N}.$$

The function provides two different `dist` for p-value estimation:

Tukey The p-values are computed from the studentized range distribution (alias `Tukey`), $\Pr \left\{t_{ij} \sqrt{2} \geq q_{k\infty\alpha} | mathrmH \right\} = \alpha$.

Chisquare The p-values are computed from the `Chisquare` distribution with $v = k - 1$ degree of freedom.

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

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

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

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

See Also

Tukey, Chisquare, p.adjust, kruskalTest, kwAllPairsDunnTest, kwAllPairsConoverTest

Examples

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

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

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

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

## Brown-Mood all-pairs median test
ans <- medianAllPairsTest(count ~ spray, data = InsectSprays)
summary(ans)
```

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

Description

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

Usage

```
kwManyOneConoverTest(x, ...)

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

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

Arguments

`x`	a numeric vector of data values, or a list of numeric data vectors.
`...`	further arguments to be passed to or from methods.
`g`	a vector or factor object giving the group for the corresponding elements of `"x"`. Ignored with a warning if `"x"` is a list.
`alternative`	the alternative hypothesis. Defaults to `two.sided`.
`p.adjust.method`	method for adjusting p values (see `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 `NA`s. Defaults to `getOption("na.action")`.

Details

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

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

See Also

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

Examples

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

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

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

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

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

```
## Many one U test
ans <- manyOneUTest(weight ~ group, data = PlantGrowth,
                        p.adjust.method = "holm")
summary(ans)

## Chen Test
ans <- chenTest(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,
  g,
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = c("single-step", p.adjust.methods),
  ...
)

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

Arguments

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

`alternative` the alternative hypothesis. Defaults to `two.sided`.

`p.adjust.method`
method for adjusting p values (see `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 `NA`s. Defaults to `getOption("na.action")`.

Details

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

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

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

See Also

pmvnorm, TDist, kruskalTest, kwManyOneConoverTest, kwManyOneNdwTest

Examples

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

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

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

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

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

## Many one U test
ans <- manyOneUTest(weight ~ group, data = PlantGrowth,
                        p.adjust.method = "holm")
summary(ans)

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

kwManyOneNdwTest *Nemenyi-Damico-Wolfe Many-to-One Rank Comparison Test*

Description

Performs Nemenyi-Damico-Wolfe non-parametric many-to-one comparison test for Kruskal-type ranked data.

Usage

```
kwManyOneNdwTest(x, ...)

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

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

Arguments

`x`	a numeric vector of data values, or a list of numeric data vectors.
`...`	further arguments to be passed to or from methods.
`g`	a vector or factor object giving the group for the corresponding elements of "x". Ignored with a warning if "x" is a list.
`alternative`	the alternative hypothesis. Defaults to `two.sided`.
`p.adjust.method`	method for adjusting p values (see `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 `NA`s. 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. $\mathrm{H}_i : \theta_0 = \theta_i$ is tested in the two-tailed case against $\mathrm{A}_i : \theta_0 \ne \theta_i, \ (1 \le i \le m)$.

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

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

Note

This function is essentially the same as `kwManyOneDunnTest`, but there is no tie correction included. Therefore, the implementation of Dunn's test is superior, when ties are present.

References

Damico, J. A., Wolfe, D. A. (1989) Extended tables of the exact distribution of a rank statistic for treatments versus control multiple comparisons in one-way layout designs, *Communications in Statistics - Theory and Methods* **18**, 3327–3353.

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

See Also

`pmvt`, `TDist`, `kruskalTest`, `kwManyOneDunnTest`, `kwManyOneConoverTest`

Examples

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

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

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

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

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

## Many one U test
ans <- manyOneUTest(weight ~ group, data = PlantGrowth,
                        p.adjust.method = "holm")
summary(ans)

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

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : \theta_1 \le \theta_2 \le \ldots \le \theta_k,\ \theta_1 < \theta_k$.

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.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

Le, C. T. (1988) A new rank test against ordered alternatives in k-sample problems, *Biometrical Journal* **30**, 87–92.

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
```

```
shanTest(x, g)
```

lsdTest *Least Significant Difference Test*

Description

Performs the least significant difference all-pairs comparisons test for normally distributed data with equal group variances.

Usage

```
lsdTest(x, ...)

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

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

## 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 `NA`s. Defaults to `getOption("na.action")`.

Details

For all-pairs comparisons in an one-factorial layout with normally distributed residuals and equal variances the least signifiant difference test can be performed after a significant ANOVA F-test. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j \quad (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Fisher's LSD all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{s_{\mathrm{in}} \left(1/n_j + 1/n_i\right)^{1/2}}, \quad (i \ne j)$$

with s_{in}^2 the within-group ANOVA variance. The null hypothesis is rejected if $|t_{ij}| > t_{v\alpha/2}$, with $v = N - k$ degree of freedom. The p-values (two-tailed) are computed from the `TDist` 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 included, 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 adustment) see function `pairwise.t.test`.

References

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

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 <- lsdTest(fit)
summary(res)
summaryGroup(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 assymptotic 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 `NA`s. 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 decresing 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.

$\mathrm{H}_i: \theta_0 = \theta_i = \ldots = \theta_k$ is tested against the alternative $\mathrm{A}_i: \theta_1 \le \ldots \theta_p \ge \theta_k$ for some p, with at least one strict inequality.

If `p = NULL` (peak unknown), the upper-tail p-value is computed via an asymptotic bootstrap permutation test.

If an integer value for `p` is given (peak known), the upper-tail p-value is computed from the standard normal distribution (`pnorm`).

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

Note

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

References

Chen, I. Y. (1991) Notes on the Mack-Wolfe and Chen-Wolfe Tests for Umbrella Alternatives, *Biom. J.* **33**, 281–290.

Mack, G. A., Wolfe, D. A. (1981) K-sample rank tests for umbrella alternatives, *J. Amer. Statist. Assoc.* **76**, 175–181.

See Also

`pnorm`, `sample`.

Examples

```
## Example from Table 6.10 of Hollander and Wolfe (1999).
## Plates with Salmonella bacteria of strain TA98 were exposed to
## various doses of Acid Red 114 (in mu g / ml).
## The data are the numbers of visible revertant colonies on 12 plates.
## Assume a peak at D333 (i.e. p = 3).
x <- c(22, 23, 35, 60, 59, 54, 98, 78, 50, 60, 82, 59, 22, 44,
  33, 23, 21, 25)
g <- as.ordered(rep(c(0, 100, 333, 1000, 3333, 10000), each=3))
plot(x ~ g)
mackWolfeTest(x=x, g=g, p=3)
```

`Mandel-h` *Mandel's h Distribution*

Description

Distribution function and quantile function for Mandel's h distribution.

Usage

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

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

Arguments

`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:
Stephen L R Ellison. (2017). metRology: Support for Metrological Applications. R package version 0.9-26-2. https://CRAN.R-project.org/package=metRology

References

Practice E 691 (2005) *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

See Also

mandelhTest

Examples

```
## 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

```
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 $P[X \leq x]$ 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:
Stephen L R Ellison. (2017). metRology: Support for Metrological Applications. R package version 0.9-26-2. https://CRAN.R-project.org/package=metRology

Note

The functions are only appropriate for balanced designs.

References

Practice E 691 (2005) *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

See Also

`mandelkTest`

`pmandelh`, `qmandelh`

Examples

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

`mandelhTest` *Mandel's h Test According to E 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 `NA`s. 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

Practice E 691 (2005) *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

See Also

qmandelh pmandelh

Examples

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

---

mandelkTest *Mandel's k Test According to E 691 ASTM*

Description

The function calculates the consistency statistics k and corresponding p-values for each group (lab) according to Practice E 691 ASTM.

Usage

```
mandelkTest(x, ...)

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

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

Arguments

`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 `NA`s. 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

Practice E 691 (2005) *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

See Also

`qmandelk` `pmandelk`

Examples

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

manyOneUTest *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 `NA`s. Defaults to `getOption("na.action")`.

Details

This functions performs Wilcoxon, Mann and Whitney's U-test for a one factorial design where each factor level is tested against one control ($m = k - 1$ tests). As the data are re-ranked for each comparison, this test is only suitable for balanced (or almost balanced) experimental designs.

For the two-tailed test and `p.adjust.method = "single-step"` the multivariate normal distribution is used for controlling Type 1 error and to calculate p-values. Otherwise, the p-values are calculated from the standard normal distribution with any latter p-adjustment as available by `p.adjust`.

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

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

See Also

`wilcox.test`, `pmvnorm`, `Normal`

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)

## Many one U test
ans <- manyOneUTest(weight ~ group, data = PlantGrowth,
                        p.adjust.method = "holm")
summary(ans)

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

medianAllPairsTest *Brown-Mood All Pairs Median Test*

Description

Performs Brown-Mood All Pairs Median Test.

Usage

```
medianAllPairsTest(x, ...)

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

```
## S3 method for class 'formula'
medianAllPairsTest(
  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 to getOption("na.action").

Details

For all-pairs comparisons in an one-factorial layout with non-normally distributed residuals Brown-Mood non-parametric Median test can be performed. A total of $m = k(k-1)/2$ hypotheses can be tested. The null hypothesis $\mathrm{H}_{ij} : \mu_i(x) = \mu_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \mu_i(x) \ne \mu_j(x), \; i \ne j$.

In this procedure the joined median is used for classification, but pairwise Pearson Chisquare-Tests are conducted. Any method as given by p.adjust.methods can be used to account for multiplicity.

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

Brown, G.W., Mood, A.M., 1951, On Median Tests for Linear Hypotheses, in: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, pp. 159–167.

See Also

`chisq.test`.

Examples

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

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

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

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

## Brown-Mood all-pairs median test
ans <- medianAllPairsTest(count ~ spray, data = InsectSprays)
summary(ans)
```

`medianTest` *Brown-Mood Median Test*

Description

Performs Brown-Mood Median Test.

Usage

```
medianTest(x, ...)

## Default S3 method:
medianTest(x, g, simulate.p.value = FALSE, B = 2000, ...)

## S3 method for class 'formula'
medianTest(
  formula,
  data,
  subset,
  na.action,
  simulate.p.value = FALSE,
  B = 2000,
  ...
)
```

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.

`simulate.p.value`
a logical indicating whether to compute p-values by Monte-Carlo simulation.

`B` an integer specifying the number of replicates used in the Monte-Carlo test.

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

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

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

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

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against the alternative, $\mathrm{H}_{\mathrm{A}} : \theta_i \ne \theta_j ~~ (i \ne j)$, with at least one unequality beeing strict.

Value

A list with class ‘htest’. For details see `chisq.test`.

References

Brown, G.W., Mood, A.M., 1951, On Median Tests for Linear Hypotheses, in: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, pp. 159–167.

See Also

chisq.test.

Examples

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

## Median Test asymptotic
medianTest(x ~ g, dat)

## Median Test with simulated p-values
set.seed(112)
medianTest(x ~ g, dat, simulate.p.value = TRUE)
```

mrrTest *Madhava Rao-Raghunath Test for Testing Treatment vs. Control*

Description

The function has implemented the nonparametric test of Madhava Rao and Raghunath (2016) for testing paired two-samples for symmetry. The null hypothesis $H : F(x, y) = F(y, x)$ is tested against the alternative $A : F(x, y) \neq F(y, x)$.

Usage

```
mrrTest(x, ...)

## Default S3 method:
mrrTest(x, y = NULL, m = NULL, ...)

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

Arguments

x	numeric vector of data values. Non-finite (e.g., infinite or missing) values will be omitted.
...	further arguments to be passed to or from methods.
y	an optional numeric vector of data values: as with x non-finite values will be omitted.
m	numeric, optional integer number, whereas $n = km$ needs to be full filled.
formula	a formula of the form `response ~ group` where `response` gives the data values and `group` a vector or factor of the corresponding groups.
data	an optional matrix or data frame (or similar: see `model.frame`) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
subset	an optional vector specifying a subset of observations to be used.
na.action	a function which indicates what should happen when the data contain `NA`s. Defaults to `getOption("na.action")`.

Details

Let X_i and $Y_i,\ i \leq n$ denote continuous variables that were observed on the same ith test item (e.g. patient) with $i = 1, \ldots n$. Let

$$U_i = X_i + Y_i \qquad V_i = X_i - Y_i$$

Let $U_{(i)}$ be the ith order statistic, $U_{(1)} \leq U_{(2)} \leq \ldots U_{(n)}$ and k the number of clusters, with the condition:

$$n = k\, m.$$

Further, let the divider denote $d_0 = -\infty$, $d_k = \infty$, and else

$$d_j = \frac{U_{(jm)} + U_{(jm+1)}}{2},\ 1 \leq j \leq k - 1$$

The two counts are

$$n_j^+ = \left\{ \begin{array}{ll} 1 & \text{if } d_{j-1} < u_i < d_j, v_i > 0 \\ 0 & \end{array} \right.$$

and

$$n_j^- = \left\{ \begin{array}{ll} 1 & \text{if } d_{j-1} < u_i < d_j, v_i \leq 0 \\ 0 & \end{array} \right.$$

The test statistic is

$$M = \sum_{j=1}^{k} \frac{\left(n_j^+ - n_j^-\right)^2}{m}$$

The exact p-values for $5 \leq n \leq 30$ are taken from an internal look-up table. The exact p-values were taken from Table 7, Appendix B of Madhava Rao and Raghunath (2016).

If `m = NULL` the function uses $n = m$ for all prime numbers, otherwise it tries to find an value for m in such a way, that for $k = n/m$ all variables are integer.

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 returns an error code if a value for `m` is provided that does not lead to an integer of the ratio $k = n/m$.

The function also returns an error code, if a tabulated value for given n, m and calculated M can not be found in the look-up table.

References

Madhava Rao, K.S., Ragunath, M. (2016) A Simple Nonparametric Test for Testing Treatment Versus Control. *J Stat Adv Theory Appl* **16**, 133–162. doi:10.18642/jsata_7100121717

Examples

```
## Madhava Rao and Raghunath (2016), p. 151
## Inulin clearance of living donors
## and recipients of their kidneys
x <- c(61.4, 63.3, 63.7, 80.0, 77.3, 84.0, 105.0)
y <- c(70.8, 89.2, 65.8, 67.1, 87.3, 85.1, 88.1)
mrrTest(x, y)

## formula method
## Student's Sleep Data
mrrTest(extra ~ group, data = sleep)
```

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.

Usage

```
MTest(x, ...)

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

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

## S3 method for class 'aov'
MTest(x, 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 `NA`s. Defaults to `getOption("na.action")`.

Details

The procedure uses the property of a simple order, $\theta'_m - \mu_m \leq \mu_j - \mu_i \leq \mu'_l - \mu_l \qquad (l \leq i \leq m \text{ and } m' \leq j \leq l')$. The null hypothesis $\mathrm{H}_{ij} : \mu_i = \mu_j$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i < \mu_j$ for any $1 \leq i < j \leq k$.

The all-pairs comparisons test statistics for a balanced design are

$$\hat{h}_{ij} = \max_{i \leq m < m' \leq j} \frac{(\bar{x}_{m'} - \bar{x}_m)}{s_{\mathrm{in}} / \sqrt{n}},$$

with $n = n_i$; $N = \sum_i^k n_i \quad (1 \leq i \leq k)$, $\bar{x}_i$ the arithmetic mean of the ith group, and s^2_{in} the within ANOVA variance. The null hypothesis is rejected, if $\hat{h} > h_{k,\alpha,v}$, with $v = N - k$ degree of freedom.

For the unbalanced case with moderate imbalance the test statistic is

$$\hat{h}_{ij} = \max_{i \leq m < m' \leq j} \frac{(\bar{x}_{m'} - \bar{x}_m)}{s_{\mathrm{in}} \left(1/n_m + 1/n_{m'}\right)^{1/2}},$$

The null hypothesis is rejected, if $\hat{h}_{ij} > h_{k,\alpha,v} / \sqrt{2}$.

The function does not return p-values. Instead the critical h-values as given in the tables of Hayter (1990) for $\alpha = 0.05$ (one-sided) are looked up according to the number of groups (k) and the degree of freedoms (v).

Value

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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

Note

The function will give a warning for the unbalanced case and returns the critical value $h_{k,\alpha,\infty}/\sqrt{2}$.

References

Hayter, A. J.(1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative, *Journal of the American Statistical Association* **85**, 778–785.

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

See Also

osrtTest, NPMTest

Examples

```
##
md <- aov(weight ~ group, PlantGrowth)
anova(md)
osrtTest(md)
MTest(md)
```

---

normalScoresAllPairsTest
Lu-Smith All-Pairs Comparison Normal Scores Test

Description

Performs Lu-Smith all-pairs comparison normal scores test.

Usage

```
normalScoresAllPairsTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`p.adjust.method` method for adjusting p values (see `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 `NA`s. 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 $\mathrm{H}_{ij} : F_i(x) = F_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : F_i(x) \ne F_j(x), \;\; i \ne j$. For `p.adjust.method = "single-step"` the Tukey's studentized range distribution is used to calculate p-values (see `Tukey`). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by `p.adjust`.

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

Lu, H., Smith, P. (1979) Distribution of normal scores statistic for nonparametric one-way analysis of variance. *Journal of the American Statistical Association* **74**, 715–722.

See Also

normalScoresTest, normalScoresManyOneTest, normOrder.

---

normalScoresManyOneTest *Lu-Smith Many-One Comparisons Normal Scores Test*

Description

Performs Lu-Smith multiple comparison normal scores test with one control.

Usage

```
normalScoresManyOneTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`alternative` the alternative hypothesis. Defaults to `two.sided`.

`p.adjust.method`
method for adjusting p values (see `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 `NA`s. Defaults to `getOption("na.action")`.

Details

For many-to-one comparisons in an one-factorial layout with non-normally distributed residuals Lu and Smith's normal scores transformation can be used prior to a many-to-one comparison test. A total of $m = k - 1$ hypotheses can be tested. The null hypothesis $\mathrm{H}_i : F_0(x) = F_i(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_i : F_0(x) \ne F_i(x), ~~ 1 \le i \le k - 1$. 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.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see `order`, `factor`).

References

Lu, H., Smith, P. (1979) Distribution of normal scores statistic for nonparametric one-way analysis of variance. *Journal of the American Statistical Association* **74**, 715–722.

See Also

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

Examples

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

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

`normalScoresTest` *Lu-Smith Normal Scores Test*

Description

Performs the Lu-Smith normal score test

Usage

```
normalScoresTest(x, ...)

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

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

Arguments

`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 `NA`s. 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 $\mathrm{H}_0 : F_1(x) = F_2(x) = \ldots = F_k(x)$ against the $\mathrm{H}_\mathrm{A} : F_i(x) \neq F_j(x)~(i \neq j)$ with at least one strict inequality. This function is basically a wrapper function to `pNormScore` of the package **SuppDists**.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

Lu, H., Smith, P. (1979) Distribution of normal scores statistic for nonparametric one-way analysis of variance. *Journal of the American Statistical Association* **74**, 715–722.

See Also

`vanWaerdenTest`, `kruskalTest`, `pNormScore`

Examples

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

`NPMTest` *All-Pairs Comparisons for Simply Ordered Mean Ranksums*

Description

Performs Nashimoto and Wright's all-pairs comparison procedure for simply ordered mean ranksums.

Usage

```
NPMTest(x, ...)

## Default S3 method:
NPMTest(
  x,
  g,
  alternative = c("greater", "less"),
  method = c("look-up", "boot", "asympt"),
  nperm = 10000,
  ...
)

## S3 method for class 'formula'
NPMTest(
  formula,
  data,
  subset,
  na.action,
  alternative = c("greater", "less"),
  method = c("look-up", "boot", "asympt"),
  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 `greater`.

`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 `NA`s. Defaults to `getOption("na.action")`.

Details

The procedure uses the property of a simple order, $\theta_m' - \theta_m \le \theta_j - \theta_i \le \theta_l' - \theta_l \qquad (l \le i \le m$ and $m' \le j \le l')$. The null hypothesis $\mathrm{H}_{ij} : \theta_i = \theta_j$ is tested against the alternative $\mathrm{A}_{ij} : \theta_i < \theta_j$ for any $1 \le i < j \le k$.

The all-pairs comparisons test statistics for a balanced design are

$$\hat{h}_{ij} = \max_{i \le m < m' \le j} \frac{\left(\bar{R}_{m'} - \bar{R}_m\right)}{\sigma_a / \sqrt{n}},$$

with $n = n_i;\ N = \sum_i^k n_i \quad (1 \le i \le k)$, $\bar{R}_i$ the mean rank for the ith group, and $\sigma_a = \sqrt{N \left(N + 1\right) / 12}$. The null hypothesis is rejected, if $h_{ij} > h_{k,\alpha,\infty}$.

For the unbalanced case with moderate imbalance the test statistic is

$$\hat{h}_{ij} = \max_{i \le m < m' \le j} \frac{\left(\bar{R}_{m'} - \bar{R}_m\right)}{\sigma_a \left(1/n_m + 1/n_{m'}\right)^{1/2}},$$

The null hypothesis is rejected, if $\hat{h}_{ij} > h_{k,\alpha,\infty} / \sqrt{2}$.

If `method = "look-up"` the function will not return p-values. Instead the critical h-values as given in the tables of Hayter (1990) for $\alpha = 0.05$ (one-sided) are looked up according to the number of groups (k) and the degree of freedoms ($v = \infty$).

If `method = "boot"` an asymetric permutation test is conducted and p-values is returned.

If `method = "asympt"` is selected the asymptotic p-value is estimated as implemented in the function `pHayStonLSA` of the package **NSM3**.

Value

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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

Either a list of class `"PMCMR"` or a list with class `"osrt"` that contains 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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

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

The function will give a warning for the unbalanced case and returns the critical value $h_{k,\alpha,\infty}/\sqrt{2}$.

Source

If `method = "asympt"` is selected, this function calls an internal probability function `pHS`. The GPL-2 code for this function was taken from `pHayStonLSA` of the the package **NSM3**:

Grant Schneider, Eric Chicken and Rachel Becvarik (2020) NSM3: Functions and Datasets to Accompany Hollander, Wolfe, and Chicken - Nonparametric Statistical Methods, Third Edition. R package version 1.15. https://CRAN.R-project.org/package=NSM3

References

Hayter, A. J.(1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative, *Journal of the American Statistical Association* **85**, 778–785.

Nashimoto, K., Wright, F.T. (2007) Nonparametric Multiple-Comparison Methods for Simply Ordered Medians. *Comput Stat Data Anal* **51**, 5068–5076.

See Also

MTest

Examples

```
## Example from Shirley (1977)
## Reaction times of mice to stimuli to their tails.
x <- 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)

## Shirley's test
## one-sided test using look-up table
shirleyWilliamsTest(x ~ g, alternative = "greater")

## Chacko's global hypothesis test for 'greater'
chackoTest(x , g)

## post-hoc test, default is standard normal distribution (NPT'-test)
summary(chaAllPairsNashimotoTest(x, g, p.adjust.method = "none"))

## same but h-distribution (NPY'-test)
chaAllPairsNashimotoTest(x, g, dist = "h")

## NPM-test
NPMTest(x, g)

## Hayter-Stone test
hayterStoneTest(x, g)

## all-pairs comparisons
hsAllPairsTest(x, g)
```

osrtTest *One-Sided Studentized Range Test*

Description

Performs Hayter’s one-sided studentized range test against an ordered alternative for normal data with equal variances.

Usage

```
osrtTest(x, ...)

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

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

## S3 method for class 'aov'
osrtTest(x, 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

Hayter's one-sided studentized range test (OSRT) can be used for testing several treatment levels with a zero control in a balanced one-factorial design with normally distributed variables that have a common variance. The null hypothesis, H: $\mu_i = \mu_j \quad (i < j)$ is tested against a simple order alternative, A: $\mu_i < \mu_j$, with at least one inequality being strict.

The test statistic is calculated as,

$$\hat{h} = \max_{1 \le i < j \le k} \frac{(\bar{x}_j - \bar{x}_i)}{s_{\mathrm{in}} / \sqrt{n}},$$

with k the number of groups, $n = n_1, n_2, \ldots, n_k$ and s_{in}^2 the within ANOVA variance. The null hypothesis is rejected, if $\hat{h} > h_{k,\alpha,v}$, with $v = N - k$ degree of freedom.

For the unbalanced case with moderate imbalance the test statistic is

$$\hat{h} = \max_{1 \le i < j \le k} \frac{(\bar{x}_j - \bar{x}_i)}{s_{\mathrm{in}} \sqrt{1/n_j + 1/n_i}},$$

The function does not return p-values. Instead the critical h-values as given in the tables of Hayter (1990) for $\alpha = 0.05$ (one-sided) are looked up according to the number of groups (k) and the degree of freedoms (v). Non tabulated values are linearly interpolated with the function approx.

Value

A list with class "osrt" that contains 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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

Note

Hayter (1990) has tabulated critical h-values for balanced designs only. For some unbalanced designs some $k = 3$ critical h-values can be found in Hayter et al. 2001. ' The function will give a warning for the unbalanced case and returns the critical value $h_{k,\alpha,v}/\sqrt{2}$.

References

Hayter, A. J.(1990) A One-Sided Studentised Range Test for Testing Against a Simple Ordered Alternative, *Journal of the American Statistical Association* **85**, 778–785.

Hayter, A.J., Miwa, T., Liu, W. (2001) Efficient Directional Inference Methodologies for the Comparisons of Three Ordered Treatment Effects. *J Japan Statist Soc* **31**, 153–174.

See Also

link{hayterStoneTest} MTest

Examples

```
##
md <- aov(weight ~ group, PlantGrowth)
anova(md)
osrtTest(md)
MTest(md)
```

pageTest *Page Rank Sum Test*

Description

Performs Page's ordered aligned rank sum test.

Usage

```
pageTest(y, ...)

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

Arguments

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

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

Page, E. B. (1963) Ordered hypotheses for multiple treatments: A significance test for linear ranks, *Journal of the American Statistical Association* **58**, 216–230.

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

See Also

friedmanTest

Examples

```
## Sachs (1997), pp. 671 ff.
## 9 reviewers (blocks)
## assigned ranks to 4 objects (groups).
data(reviewers)
## See Sachs (1997) p. 677
pageTest(reviewers, alternative = "greater")
```

Pentosan *Pentosan Dataset*

Description

A benchmark dataset of an interlaboratory study for determining the precision of a test method on several levels of the material Pentosan.

Format

A data frame with 189 obs. of 3 variables:

value numeric, test result (no unit specified)

lab factor, identifier of the lab (1–7)

material factor, identifier of the level of the material (A–I)

Source

Tab. 8, Practice E 691, 2005, *Standard Practice for Conducting an Interlaboratory Study to Determine the Precision of a Test Method*, ASTM International.

plot.mandel *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

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

power.dunnett.test *Power Calculations for Balanced Dunnett's Many-to-One Comparison Test*

Description

Compute average per-pair power of Dunnetts's multiple comparison test with one control.

Usage

```
power.dunnett.test(n, groups, delta, within.var, sig.level = 0.05)
```

Arguments

n	Number of observations (per group)
groups	Number of groups (including control)
delta	true difference in means
within.var	Within group variance
sig.level	Significance level (Type I error probability)

Details

The function has implemented the following Eq. to estimate average per-pair power for two-sided tests:

$$1 - \beta = 1 - t(T_{\alpha\rho v}, v, \mathrm{ncp}) + t(-T_{\alpha\rho v}, v, \mathrm{ncp}),$$

with $T_{\alpha\rho v}$ the two-sided α quantile of the multivariate t-distribution, with $v = k(n-1)$ degree of freedom, k the number of groups and correlation matrix $\rho_{ij} = 0.5 \; (i \neq j)$.

The non-centrality parameter for the non-central student t-distribution is

$$\mathrm{ncp} = |\Delta| / \sqrt{s^2_{\mathrm{in}} \; 2/n}.$$

Value

Object of class 'power.htest', a list of the arguments (including the computed one) augmented with method and note elements.

Note

The results for power are seed depending.

Source

The Eqs. were taken from Lecture 5, *Determining Sample Size*, Statistics 514, Fall 2015, Purdue University, IN, USA.

See Also

TDist qmvt powerMCTests

Examples

```
set.seed(113)
power.dunnett.test(n = 9, groups = 5, delta = 30,
 within.var = 333.7)

## compare with t-test, bonferroni corrected
power.t.test(n = 9, delta = 30, sd = sqrt(333.7),
sig.level = 0.05 / 4)

## Not run:
## asymptotic Monte-Carlo power analysis
 set.seed(113)
 powerMCTests(mu = c(rep(0,4), 30), n = 9,
 parms = list(mean = 0, sd = sqrt(333.7)),
 test = "dunnettTest", alternative = "two.sided")

## End(Not run)
```

power.tukey.test *Power Calculations for Balanced Tukey's Multiple Comparison Test*

Description

Compute average per-pair power of Tukey's test for multiple comparison of means.

Usage

```
power.tukey.test(n, groups, delta, within.var, sig.level = 0.05)
```

Arguments

`n`	number of observations (per group)
`groups`	number of groups
`delta`	true difference in means
`within.var`	within group variance
`sig.level`	significance level (Type I error probability)

Details

The function has implemented the following Eq. to estimate average per-pair power for two-sided tests:

$$1 - \beta = 1 - t(q_{\alpha v k}/\sqrt{2}, v, \mathrm{ncp}) + t(-q_{\alpha v k}/\sqrt{2}, v, \mathrm{ncp}),$$

with $q_{\alpha v k}$ the upper α quantile of the studentised range distribution, with $v = k(n-1)$ degree of freedom and k the number of groups; and $t(.\ \mathrm{ncp})$ the probability function of the non-central student t-distribution with non-centrality parameter

$$\mathrm{ncp} = |\Delta| / \sqrt{s^2_{\mathrm{in}}\ 2/n}.$$

Value

Object of class 'power.htest', a list of the arguments (including the computed one) augmented with `method` and `note` elements.

Source

The Eqs. were taken from Lecture 5, *Determining Sample Size*, Statistics 514, Fall 2015, Purdue University, IN, USA.

See Also

`TDist` `Tukey` `powerMCTests`

Examples

```
power.tukey.test(n = 11, groups = 5, delta = 30,
 within.var = 333.7)

## compare with t-test, Bonferroni-correction
power.t.test(n = 11, delta = 30, sd = sqrt(333.7),
sig.level = 0.05 / 10)

## Not run:
powerMCTests(mu = c(rep(0,4), 30), n = 11,
 parms = list(mean = 0,sd = sqrt(333.7)),
 test = "tukeyTest")

## End(Not run)
```

`power.williams.test` *Power calculations for minimum detectable difference of the Williams' test*

Description

Compute the power of a Williams' test, or determine parameters to obtain a target power.

Usage

```
power.williams.test(n = NULL, k, delta, sd = 1, power = NULL, ...)
```

Arguments

`n` number of observations (per group).

`k` number of treatment groups.

`delta` clinically meaningful minimal difference (between a treatment group and control).

`sd` common standard deviation.

`power` power of test (1 minus Type II error probability).

`...` further arguments, currently ignored.

Details

Exactly one of the parameters `n` or `power` must be passed as `NULL`, and that parameter is determined from the others.

The function has implemented the following Eq. in order to estimate power (Chow et al. 2008):

$$1 - \beta = 1 - \Phi\left(T_{K\alpha v} - |\Delta|/\sigma\sqrt{2/n}\right)$$

with $|\Delta|$ the clinically meaningful minimal difference, $T_{K\alpha v}$ the critical Williams' t-statistic for $\alpha = 0.05$, $v = \infty$ degree of freedom and Φ the probability function of the standard normal function.

The required sample size (balanced design) is estimated based on the expression as given by the PASS manual, p. 595-2:

$$n = 2 \sigma^2 ~ \left(T_{K\alpha v} + z_{\beta}\right)^2 ~ / \Delta^2$$

Value

Object of class 'power.htest', a list of the arguments (including the computed one) augmented with method and note elements.

Note

The current function calculates power for sig.level = 0.05 significance level (Type I error probability) only (one-sided test).

References

Chow, S.-C., Shao, J., Wan, H., 2008, *Sample Size Calculations in Clinical Research*, 2nd ed, Chapman & Hall/CRC: Boca Raton, FL.

See Also

optimise williamsTest

Examples

```
## Chow et al. 2008, p. 288 depicts 53 (rounded),
## better use ceiling for rounding
power.williams.test(power = 0.8, k = 3, delta = 11, sd = 22)
power.williams.test(n = 54, k = 3, delta = 11, sd = 22)

## PASS manual example:
## up-rounded n values are:
## 116, 52, 29, 14, 8 and 5
## according to PASS manual, p. 595-5
D <- c(10, 15, 20, 30, 40, 50)
y <- sapply(D, function(delta) {
 power.williams.test(power = 0.9, k = 4, delta = delta, sd = 25)$n
 })
ceiling(y)

## Not run:
 ## compare with power.t.test
 ## and bonferroni correction
 power.t.test(power = 0.9, delta = 50, sd = 25,
 sig.level = 0.05 / 4, alternative = "one.sided")

## End(Not run)
```

powerMCTests *Power Simulation for One-Factorial All-Pairs and Many-To-One Comparison Tests*

Description

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

Usage

```
powerMCTests(
  mu,
  n = 10,
  errfn = c("Normal", "Lognormal", "Exponential", "Chisquare", "TDist", "Cauchy",
    "Weibull"),
  parms = list(mean = 0, sd = 1),
  test = c("kwManyOneConoverTest", "kwManyOneDunnTest", "kwManyOneNdwTest",
    "vanWaerdenManyOneTest", "normalScoresManyOneTest", "dunnettTest",
    "tamhaneDunnettTest", "ManyOneUTest", "chenTest", "kwAllPairsNemenyiTest",
    "kwAllPairsDunnTest", "kwAllPairsConoverTest", "normalScoresAllPairsTest",
    "vanWaerdenAllPairsTest", "dscfAllPairsTest", "gamesHowellTest", "lsdTest",
   "scheffeTest", "tamhaneT2Test", "tukeyTest", "dunnettT3Test", "pairwise.t.test",
    "pairwise.wilcox.test", "adManyOneTest", "adAllPairsTest", "bwsManyOneTest",

    "bwsAllPairsTest", "welchManyOneTTest"),
  alternative = c("two.sided", "greater", "less"),
  p.adjust.method = c("single-step", p.adjust.methods),
  alpha = 0.05,
  FWER = TRUE,
  replicates = 1000
)
```

Arguments

`mu` numeric vector of group means.

`n` number of replicates per group. If `n` is a scalar, then a balanced design is assumed. Otherwise, `n` must be a vector of same length as `mu`.

`errfn` the error function. Defaults to `"Normal"`.

`parms` a list that denotes the arguments for the error function. Defaults to `list(mean=0, sd=1)`.

`test` the multiple comparison test for which the power analysis is to be performed. Defaults to `"kwManyOneConoverTest"`.

`alternative` the alternative hypothesis. Defaults to `"two.sided"`, ignored if the selected error function does not use this argument.

`p.adjust.method`

method for adjusting p values (see `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 ϵ_{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

```
## Not run:
mu <- c(0, 0, 1, 2)
n <- c(5, 4, 5, 5)
set.seed(100)
powerMCTests(mu, n, errfn="Normal",
 parms=list(mean=0, sd=1),
 test="dunnettTest", replicates=1E4)

powerMCTests(mu, n, errfn="Normal",
 parms=list(mean=0, sd=1),
 test="kwManyOneDunnTest", p.adjust.method = "bonferroni",
 replicates=1E4)


## End(Not run)
```

`powerOneWayTests` *Power Simulation for One-Factorial Single Hypothesis Tests*

Description

Performs power simulation for one-factorial single hypothesis tests.

Usage

```
powerOneWayTests(
  mu,
  n = 10,
  errfn = c("Normal", "Lognormal", "Exponential", "Chisquare", "TDist", "Cauchy",
    "Weibull"),
  parms = list(mean = 0, sd = 1),
  test = c("kruskalTest", "leTest", "vanWaerdenTest", "normalScoresTest", "spearmanTest",
   "cuzickTest", "jonckheereTest", "johnsonTest", "oneway.test", "adKSampleTest",
   "bwsKSampleTest", "bwsTrendTest", "mackWolfeTest", "chackoTest", "flignerWolfeTest"),
  alternative = c("two.sided", "greater", "less"),
  var.equal = TRUE,
  dist = NULL,
  alpha = 0.05,
  FWER = TRUE,
  replicates = 1000,
  p = NULL
)
```

Arguments

`mu` numeric vector of group means.

`n` number of replicates per group. If `n` is a scalar, then a balanced design is assumed. Otherwise, `n` must be a vector of same length as `mu`.

`errfn` the error function. Defaults to `"Normal"`.

`parms` a list that denotes the arguments for the error function. Defaults to `list(mean=0, sd=1)`.

`test` the test for which the power analysis is to be performed. Defaults to `"kwManyOneConoverTest"`.

`alternative` the alternative hypothesis. Defaults to `"two.sided"`, ignored if the selected error function does not use this argument.

`var.equal` a logical variable indicating whether to treat the variances in the samples as equal. `"TRUE"`, then a simple F test for the equality of means in a one-way analysis of variance is performed. If `"FALSE"`, an approximate method of Welch (1951) is used, which generalizes the commonly known 2-sample Welch test to the case of arbitrarily many samples. Defaults to `"TRUE"`; only relevant, if `test = "oneway.test"`, otherwise ignored.

`dist` the test distribution. Only relevant for `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 ϵ_{ij} based on the given error function and the corresponding parameters. Then the specified test is performed. Finally, Type I and Type II error rates are calculated.

Value

An object with class powerOneWayPMCMR.

See Also

powerMCTests, pwr.anova.test, power.anova.test

Examples

```
## Not run:
set.seed(12)
mu <- c(0, 0, 1, 2)
n <- c(5, 4, 5, 5)
parms <- list(mean=0, sd=1)
powerOneWayTests(mu, n, parms, test = "cuzickTest",
alternative = "two.sided", replicates = 1E4)

## Compare power estimation for
## one-way ANOVA with balanced design
## as given by functions
## power.anova.test, pwr.anova.test
## and powerOneWayTest

groupmeans <- c(120, 130, 140, 150)
SEsq <- 500  # within-variance
n <- 10
k <- length(groupmeans)
df <- n * k - k
SSQ.E <- SEsq * df
SSQ.A <- n * var(groupmeans) * (k - 1)
sd.errfn <- sqrt(SSQ.E / (n * k - 1))
R2 <- c("R-squared" = SSQ.A / (SSQ.A + SSQ.E))
cohensf <- sqrt(R2 / (1 - R2))
names(cohensf) <- "Cohens f"

## R stats power function
power.anova.test(groups = k,
                 between.var = var(groupmeans),
                 within.var = SEsq,
                 n = n)
```

```
## pwr power function
pwr.anova.test(k = k, n = n, f = cohensf, sig.level=0.05)

## this Monte-Carlo based estimation
set.seed(200)
powerOneWayTests(mu = groupmeans,
                 n = n,
                 parms = list(mean=0, sd=sd.errfn),
                 test = "oneway.test",
                 var.equal = TRUE,
                 replicates = 5E3)

## Compare with effect sizes
R2
cohensf


## End(Not run)
```

`print.gesdTest` *gesdTest Printing*

Description

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

Usage

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

Arguments

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

`print.mandel` *Mandel Printing*

Description

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

Usage

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

Arguments

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

See Also

mandelhTest, mandelkTest

---

print.osrt *osrt Printing*

Description

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

Usage

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

Arguments

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

See Also

summary.osrt

print.PMCMR *PMCMR Printing*

Description

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

Usage

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

Arguments

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

print.powerOneWayPMCMR *PowerOneWayPMCMR Printing*

Description

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

Usage

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

Arguments

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

print.powerPMCMR *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

```
## 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.trendPMCMR *trendPMCMR Printing*

Description

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

Usage

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

Arguments

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

qDunnett *Dunnett Distribution*

Description

Distribution function and quantile function for the distribution of Dunnett's many-to-one comparisons test.

Usage

```
qDunnett(p, n0, n)

pDunnett(q, n0, n, lower.tail = TRUE)
```

Arguments

`p`	vector of probabilities.
`n0`	sample size for control group.
`n`	vector of sample sizes for treatment groups.
`q`	vector of quantiles.
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.

Details

Dunnett's distribution is a special case of the multivariate t distribution.

Let the total sample size be $N = n_0 + \sum_i^m n_i$, with m the number of treatment groups, than the quantile $T_{mv\rho\alpha}$ is calculated with $v = N - k$ degree of freedom and the correlation ρ

$$\rho_{ij} = \sqrt{\frac{n_i n_j}{\left(n_i + n_0\right)\left(n_j + n_0\right)}} ~~ (i \neq j).$$

The functions determines m via the length of the input vector `n`.

Quantiles and p-values are computed with the functions of the package **mvtnorm**.

Value

`pDunnett` gives the distribution function and `qDunnett` gives its inverse, the quantile function.

Note

The results are seed depending.

See Also

`qmvt` `pmvt` `dunnettTest`

Examples

```
## Table gives 2.34 for df = 6, m = 2, one-sided
set.seed(112)
qval <- qDunnett(p = 0.05, n0 = 3, n = rep(3,2))
round(qval, 2)
set.seed(112)
pDunnett(qval, n0=3, n = rep(3,2), lower.tail = FALSE)

## Table gives 2.65 for df = 20, m = 4, two-sided
set.seed(112)
qval <- qDunnett(p = 0.05/2, n0 = 5, n = rep(5,4))
round(qval, 2)
set.seed(112)
2 * pDunnett(qval, n0= 5, n = rep(5,4), lower.tail= FALSE)
```

---

qPCR *qPCR Curve Analysis Methods*

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

5PSM a numeric vector

DART a numeric vector

FPLM a numeric vector

LRE_Emax a numeric vector

FPK_PCR a numeric vector

Source

Data were taken from Table 2 of Ruijter et al. (2013, p. 38). See also Eisinga et al. (2017, pp. 14–15).

References

Eisinga, R., Heskes, T., Pelzer, B., Te Grotenhuis, M. (2017) Exact p-values for pairwise comparison of Friedman rank sums, with application to comparing classifiers. *BMC Bioinformatics*, 18:68.

Ruijter, J. M. et al. (2013) Evaluation of qPCR curve analysis methods for reliable biomarker discovery: Bias, resolution, precision, and implications, *Methods* **59**, 32–46.

`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 $\mathrm{H}_{ij} : \theta_i = \theta_j$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \theta_i \neq \theta_j, ~~ i \neq j$.

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

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

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

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

See Also

`quade.test`, `friedmanTest`

Examples

```
## Sachs, 1997, p. 675
## Six persons (block) received six different diuretics
## (A to F, treatment).
## The responses are the Na-concentration (mval)
## in the urine measured 2 hours after each treatment.
##
```

```
y <- matrix(c(
3.88, 5.64, 5.76, 4.25, 5.91, 4.33, 30.58, 30.14, 16.92,
23.19, 26.74, 10.91, 25.24, 33.52, 25.45, 18.85, 20.45,
26.67, 4.44, 7.94, 4.04, 4.4, 4.23, 4.36, 29.41, 30.72,
32.92, 28.23, 23.35, 12, 38.87, 33.12, 39.15, 28.06, 38.23,
26.65),nrow=6, ncol=6,
dimnames=list(1:6, LETTERS[1:6]))
print(y)

## Global test
quade.test(y)

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

reviewers *Reviewers*

Description

9 reviewers (blocks) assigned ranks to 4 objects (groups).

Format

The format is a 9 x 4 Matrix with Friedman type rankings:

rows reviewers, 1, 2, . . . , 9

columns groups, A, B, . . . , D

Source

Sachs (1997), p. 671 ff.

References

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

Examples

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

scheffeTest *Scheffe's Test*

Description

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

Usage

```
scheffeTest(x, ...)

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

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

## 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 `NA`s. 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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j \quad (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Scheffe's all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{s_{\mathrm{in}} \left(1/n_j + 1/n_i\right)^{1/2}}, ~~ (i \neq j)$$

with s_{in}^2 the within-group ANOVA variance. The null hypothesis is rejected if $t_{ij}^2 > F_{v_1 v_2 \alpha}$, with $v_1 = k - 1, ~ v_2 = N - k$ degree of freedom. The p-values are computed from the FDist 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

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

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

Scheffe, H. (1953) A Method for Judging all Contrasts in the Analysis of Variance, *Biometrika* **40**, 87–110.

See Also

FDist, tukeyTest

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 <- scheffeTest(fit)
summary(res)
summaryGroup(res)
```

shanTest *Testing against Ordered Alternatives (Shan-Young-Kang Test)*

Description

Performs the Shan-Young-Kang test for testing against ordered alternatives.

Usage

```
shanTest(x, ...)

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

## S3 method for class 'formula'
shanTest(
  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 `NA`s. Defaults to `getOption("na.action")`.

Details

The null hypothesis, $\mathrm{H}_0 : \theta_1 = \theta_2 = \ldots = \theta_k$ is tested against a simple order hypothesis, $\mathrm{H}_\mathrm{A} : \theta_1 \le \theta_2 \le \ldots \le \theta_k,\ \theta_1 < \theta_k$.

Let R_{ij} be the rank of X_{ij}, where X_{ij} is jointly ranked from $\left\{1, 2, \ldots, N\right\}, ~~ N = \sum_{i=1}^k n_i$, the the test statistic is

$$S = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} D_{ij},$$

with

$$D_{ij} = \sum_{l=1}^{n_i} \sum_{m=1}^{n_j} \left(R_{jm} - R_{il}\right) \mathrm{I}\left(X_{jm} > X_{il}\right),$$

where

$$\mathrm{I}(u) = \left\{ \begin{array}{lll} 1, & & \forall \, u > 0 \\ 0, & & \forall \, u \le 0 \end{array} \right. .$$

The test statistic is asymptotically normal distributed:

$$z = \frac{S - \mu_{\mathrm{S}}}{\sqrt{s^2_{\mathrm{S}}}}$$

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.

Note

The variance estimation (see Theorem 2.1, Shan et al. 2014) can become negative for certain combinations of $N, ~n_i, ~k \qquad (1 \le i \le k)$. In these cases the function will return a warning and the returned p-value will be NaN.

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see order, factor).

References

Shan, G., Young, D., Kang, L. (2014) A New Powerful Nonparametric Rank Test for Ordered Alternative Problem. PLOS ONE 9, e112924. https://doi.org/10.1371/journal.pone.0112924

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
shanTest(x, g)
```

shirleyWilliamsTest *Shirley-Williams Test*

Description

Performs Shirley's nonparametric equivalent of William's test for contrasting increasing dose levels of a treatment.

Usage

```
shirleyWilliamsTest(x, ...)

## Default S3 method:
shirleyWilliamsTest(
  x,
  g,
  alternative = c("two.sided", "greater", "less"),
  method = c("look-up", "boot"),
  nperm = 10000,
  ...
)

## 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 `NA`s. Defaults to `getOption("na.action")`.

Details

The Shirley-William test is a non-parametric step-down trend test for testing several treatment levels with a zero control. Let there be k groups including the control and let the zero dose level be indicated with $i = 0$ and the highest dose level with $i = m$, then the following `m = k - 1` hypotheses are tested:

$$\begin{array}{ll}
\mathrm{H}_{m}: \theta_0 = \theta_1 = \ldots = \theta_m, & \mathrm{A}_{m} = \theta_0 \le \theta_1 \le \ldots \theta_m, \theta_0 < \theta_m \\
\mathrm{H}_{m-1}: \theta_0 = \theta_1 = \ldots = \theta_{m-1}, & \mathrm{A}_{m-1} = \theta_0 \le \theta_1 \le \ldots \theta_{m-1}, \theta_0 < \theta_{m-1} \\
\vdots & \vdots \\
\mathrm{H}_{1}: \theta_0 = \theta_1, & \mathrm{A}_{1} = \theta_0 < \theta_1
\end{array}$$

Let R_{ij} be the rank of X_{ij}, where X_{ij} is jointly ranked from $\left\{1, 2, \ldots, N\right\}, \;\; N = \sum_{i=1}^k n_i$, then the test statistic is

$$t_i = \frac{\max_{1 \le u \le i} \left(\sum_{j=u}^i n_j \bar{R}_j / \sum_{j=u}^i n_j \right) - \bar{R}_0}{\sigma_{R_i} \sqrt{1/n_i + 1/n_0}},$$

with expected variance of

$$\sigma^2_{R_i} = N_i \left(N_i + 1\right) / 12 - T_i,$$

where $N_i = n_0 + n_1 + n_2 + \ldots + n_i$ and T_i the ties for the i-th comparison is given by

$$T_i = \sum_{j=1}^{i} \frac{t_j^3 - t_j}{12 \left(N_i - 1\right)}.$$

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), i.e. the data are re-ranked for each of the i-th comparison.

If `method = "look-up"` is selected, the function does not return p-values. Instead the critical $t'_{i,v,\alpha}$-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 β.

Non tabulated values are linearly interpolated with the function `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 (`Normal`).

If `method = "boot"`, the p-values are estimated through an assymptotic boot-strap method. The p-values for H_1 are calculated from the t distribution with infinite degree of freedom.

Value

Either a list with class `"osrt"` or a list with class `"PMCMR"`.

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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

dist a string that denotes the test distribution.

There are print and summary methods available.

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

For `method = "look-up"`, only tests on the level of $\alpha = 0.05$ can be performed for alternative hypotheses less or greater.

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

References

Shirley, E., (1977) Nonparametric Equivalent of Williams Test for Contrasting Increasing Dose Levels of a Treatment, *Biometrics* **33**, 386–389.

Williams, D. A. (1986) Note on Shirley's nonparametric test for comparing several dose levels with a zero-dose control, *Biometrics* **42**, 183–186.

See Also

`williamsTest`

Examples

```
## Example from Shirley (1977)
## Reaction times of mice to stimuli to their tails.
x <- 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)

## Shirley's test
## one-sided test using look-up table
shirleyWilliamsTest(x ~ g, alternative = "greater")

## Chacko's global hypothesis test for 'greater'
chackoTest(x , g)

## post-hoc test, default is standard normal distribution (NPT'-test)
summary(chaAllPairsNashimotoTest(x, g, p.adjust.method = "none"))

## same but h-distribution (NPY'-test)
chaAllPairsNashimotoTest(x, g, dist = "h")

## NPM-test
NPMTest(x, g)

## Hayter-Stone test
hayterStoneTest(x, g)

## all-pairs comparisons
hsAllPairsTest(x, g)
```

---

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 `NA`s. Defaults to `getOption("na.action")`.

Details

Let x and y denote two identically and independently distributed variables of at least ordinal scale. Further, let θ, and λ 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

The algorithm for the Siegel-Tukey ranks was taken from the code of Daniel Malter. See also the blog from Tal Galili (02/2010, https://www.r-statistics.com/2010/02/siegel-tukey-a-non-parametric-test-fo accessed 2018-08-05).

References

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

Siegel, S., Tukey, J. W. (1960), A nonparametric sum of ranks procedure for relative spread in unpaired samples, *Journal of the American Statistical Association* **55**, 429–455.

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 *Skillings-Mack Test*

Description

Performs Skillings-Mack rank sum test for partially balanced incomplete block designs or partially balanced random block designs. The null hypothesis $\mathrm{H}_0 : \theta_i = \theta_j \quad (i \neq j)$ is tested against the alternative $\mathrm{H}_{\mathrm{A}} : \theta_i \neq \theta_j$, with at least one inequality beeing 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 assymptotically chi-squared distributed with df = k - 1 degrees of freedom.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

Note

The input vector/matrix `'y'` must contain `NA`.

References

Skillings, J. H., Mack, G.A. (1981) On the use of a Friedman-type statistic in balanced and unbalanced block designs, *Technometrics* **23**, 171–177.

See Also

`friedmanTest`, `durbinTest`

Examples

```
## Example from Hollander and Wolfe 1999,
## originally appeared in Brady 1969.
x <- cbind(c(3,1,5,2,0,0,0,0),
           c(5,3,4,NA,2,2,3,2),
           c(15,18,21,6,17,10,8,13))
colnames(x) <- c("R", "A", "B")
```

```
rownames(x) <- 1:8
skillingsMackTest(x)

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

snkTest *Student-Newman-Keuls Test*

Description

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

Usage

```
snkTest(x, ...)

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

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

## S3 method for class 'aov'
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 `NA`s. 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 $\mathrm{H}_{ij} : \mu_i(x) = \mu_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij} : \mu_i(x) \ne \mu_j(x), \;\; i \ne j$.

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

Keuls, M. (1952) The use of the "studentized range" in connection with an analysis of variance, *Euphytica* **1**, 112–122.

Newman, D. (1939) The distribution of range in samples from a normal population, expressed in terms of an independent estimate of standard deviation, *Biometrika* **31**, 20–30.

Student (1927) Errors of routine analysis, *Biometrika* **19**, 151–164.

See Also

`Tukey`, `TukeyHSD` `tukeyTest`

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 <- snkTest(fit)
summary(res)
summaryGroup(res)
```

`spearmanTest` *Testing against Ordered Alternatives (Spearman Test)*

Description

Performs a Spearman type test for testing against ordered alternatives.

Usage

```
spearmanTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`alternative` the alternative hypothesis. Defaults to `"two.sided"`.

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

`data` an optional matrix or data frame (or similar: see `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

A one factorial design for dose finding comprises an ordered factor, .e. treatment with increasing treatment levels. The basic idea is to correlate the ranks R_{ij} with the increasing order number $1 \le i \le k$ of the treatment levels (Kloke and McKean 2015). More precisely, R_{ij} is correlated with the expected mid-value ranks under the assumption of strictly increasing median responses. Let the expected mid-value rank of the first group denote $E_1 = \left(n_1 + 1\right)/2$. The following expected mid-value ranks are $E_j = n_{j-1} + \left(n_j + 1\right)/2$ for $2 \le j \le k$. The corresponding number of tied values for the ith group is n_i. # The sum of squared residuals is $D^2 = \sum_{i=1}^k \sum_{j=1}^{n_i} \left(R_{ij} - E_i\right)^2$. Consequently, Spearman's rank correlation coefficient can be calculated as:

$$r_{\mathrm{S}} = \frac{6D^2}{\left(N^3 - N\right) - C},$$

with

$$C = 1/2 - \sum_{c=1}^{r} \left(t_c^3 - t_c\right) + 1/2 - \sum_{i=1}^{k} \left(n_i^3 - n_i\right)$$

and t_c the number of ties of the cth group of ties. Spearman's rank correlation coefficient can be tested for significance with a t-test. For a one-tailed test the null hypothesis of $r_{\mathrm{S}} \le 0$ is rejected and the alternative $r_{\mathrm{S}} > 0$ is accepted if

$$r_{\mathrm{S}} \sqrt{\frac{(n-2)}{(1-r_{\mathrm{S}})}} > t_{v,1-\alpha},$$

with $v = n - 2$ degree 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

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see order, factor).

References

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

See Also

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

Examples

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

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

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

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

## Jonckheere-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)

## Fligner-Wolfe test
flignerWolfeTest(x, g)

## Shan-Young-Kang test
```

```
shanTest(x, g)
```

steelsKSampleTest *Steel's k-Treatments vs. Control Test*

Description

Performs the non-parametric Steel's test for simultaneously testing k-treatments vs. one control.

Usage

```
steelsKSampleTest(x, ...)

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

## S3 method for class 'formula'
steelsKSampleTest(
  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 `NA`s. Defaults to `getOption("na.action")`.

Details

It tests $H: F(i) = F(0),\ i \leq k$, against $A: F(i) > F(0)$ (greater) with at least one inequality being strict.

The function is a wrapper function that calls `Steel.test` of the package **kSamples** with argument `method = "asymptotic"`.

Value

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

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

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

statistic the estimated quantile of the test statistic.

p.value the p-value for the test.

parameter the parameters of the test statistic, if any.

alternative a character string describing the alternative hypothesis.

estimates the estimates, if any.

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

References

Scholz, F. and Zhu, A. (2019). kSamples: K-Sample Rank Tests and their Combinations. R package version 1.2-9. https://CRAN.R-project.org/package=kSamples

Steel, R. G. D. (1959) A Multiple Comparison Rank Sum Test: Treatments Versus Control, *Biometrics* **15**, 560–572.

See Also

`Steel.test`, `flignerWolfeTest`

Examples

```
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
110, 125, 143, 148, 151,
136, 139, 149, 160, 174)
g <- gl(3,5)
levels(g) <- c("0", "I", "II")

## Steel's Test
steelsKSampleTest(x ~ g, alternative = "greater")


## 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, 23, 26, 0, 25, 26, 27,
18, 22, 13, 13, 23, 22, 20, 22, 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%
steelsKSampleTest(x ~ g, subset = g != "50%", alternative = "less")
```

---

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

```
steelTest(x, ...)

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

## S3 method for class 'formula'
steelTest(
  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 `NA`s. Defaults to `getOption("na.action")`.

Details

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. $\mathrm{H}_i : \theta_0 = \theta_i$ is tested in the one-tailed case (less) against $\mathrm{A}_i : \theta_0 > \theta_i, \quad (1 \le i \le k)$.

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 $R_i \le R$. 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 `"osrt"` that contains 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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

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).

USEPA (2002) *Short-term Methods for Estimating the Chronic Toxicity of Effluents and Receiving Waters to Freshwater Organisms*, 4th edition, EPA-821-R-02-013.

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

Steel, R. G. D. (1959) A multiple comparison rank sum test: treatments versus control, *Biometrics* **15**, 560–572.

See Also

wilcox.test, pairwise.wilcox.test, manyOneUTest, flignerWolfeTest, shirleyWilliamsTest, kwManyOneDunnTest, kwManyOneNdwTest, kwManyOneConoverTest, print.osrt, summary.osrt

Examples

```
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
110, 125, 143, 148, 151,
136, 139, 149, 160, 174)
g <- gl(3,5)
levels(g) <- c("0", "I", "II")

## 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, 23, 26, 0, 25, 26, 27,
18, 22, 13, 13, 23, 22, 20, 22, 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")
```

stepDownTrendTest *Step Down Trend Tests*

Description

Performs step-down trend test procedures for monotone responses to detect NOEC (LOEC) according to OECD (2006).

Usage

```
stepDownTrendTest(x, ...)

## Default S3 method:
stepDownTrendTest(
  x,
  g,
 test = c("leTest", "spearmanTest", "jonckheereTest", "cuzickTest", "chackoTest",
    "johnsonTest"),
  alternative = c("two.sided", "greater", "less"),
  continuity = FALSE,
  ...
)

## S3 method for class 'formula'
stepDownTrendTest(
  formula,
  data,
  subset,
  na.action,
 test = c("leTest", "spearmanTest", "jonckheereTest", "cuzickTest", "chackoTest",
    "johnsonTest"),
  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.
`test`	the trend test that shall be performed. Defaults to `"leTest"`.
`alternative`	the alternative hypothesis. Defaults to `"two.sided"`.
`continuity`	logical indicator whether a continuity correction shall be performed. Only relevant for `"jonckheereTest"`. 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 `NA`s. Defaults to `getOption("na.action")`.

Details

According to OECD 2006 one can perform a test for trend on responses from all dose groups including the control. If the trend test is significant at the 0.05 level, the high dose group is omitted, and the trend statistic with the remaining dose groups is re-compute The procedure is continued until the trend test is first non-significant at the 0.05 level, then stop.

The NOEC is the highest dose remaining at this stage. If this test is significant when only the lowest dose and control remain, then a NOEC cannot be established from the data.

Value

A list with class "trendPMCMR" 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.

dist a string that denotes the test distribution.

Note

Factor labels for g must be assigned in such a way, that they can be increasingly ordered from zero-dose control to the highest dose level, e.g. integers {0, 1, 2, ..., k} or letters {a, b, c, ...}. Otherwise the function may not select the correct values for intended zero-dose control.

It is safer, to i) label the factor levels as given above, and to ii) sort the data according to increasing dose-levels prior to call the function (see order, factor).

References

OECD (2006) *Current Approaches in the Statistical Analysis of Ecotoxicity Data: A Guidance to Application*, OECD Series on Testing and Assessment **52**, Paris: Organisation for Econonomic Co-operation and Development.

See Also

leTest, jonckheereTest, spearmanTest, cuzickTest, chackoTest, johnsonTest

Examples

```
res <- stepDownTrendTest(Y ~ DOSE, data = trout,
                         test = "jonckheereTest",
                         alternative = "less")
## print method
res
## summary method
summary(res)
```

summary.gesdTest *Summarize an gesdTest Object*

Description

Summarize an object of class *gesdTest*.

Usage

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

Arguments

`object`	an object of class `"gesdTest"`.
`...`	further arguments. Currenly ignored.

summary.mandel *Object Summary for class* `"mandel"`

Description

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

Usage

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

Arguments

`object`	an object of class `"mandel"` for which a summary is desired.
`...`	further arguments. Currently ignored.

See Also

`mandelhTest`, `mandelkTest`

summary.osrt *Summarize an osrt Object*

Description

Summarize an object of class *osrt*.

Usage

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

Arguments

`object`	an object of class `"osrt"`.
`...`	further arguments. Currenly ignored.

See Also

`print.osrt`.

summary.PMCMR *Summarize an PMCMR Object*

Description

Summarize an object of class *PMCMR*.

Usage

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

Arguments

`object`	an object of class `"PMCMR"`.
`...`	further arguments. Currenly 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.trendPMCMR` *Summarize an trendPMCMR Object*

Description

Summarize an object of class *trendPMCMR*.

Usage

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

Arguments

`object`	an object of class "`trendPMCMR`".
`...`	further arguments. Currenly 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.trendPMCMR`

`summaryGroup` *Grouped Summary of an PMCMR Object*

Description

Performes a grouped summary on an PMCMR object.

Usage

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

Arguments

`x`	an object of class "`PMCMR`".
`alpha`	the selected alpha-level. Defaults to 0.05.
`...`	further arguments. Currently ignored.

Value

Provides summary statistics for each factor level and a letter symbol, whereas different letters indicate significant differences between factor levels based on the selected level of alpha.

See Also

summary.PMCMR

`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. Let X_{0j} denote a continuous random variable with the j-the realization of the control group ($1 \leq j \leq n_0$) and X_{ij} the j-the realization in the i-th treatment group ($1 \leq i \leq k$). Furthermore, the total sample size is $N = n_0 + \sum_{i=1}^k n_i$. A total of $m = k$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_i : \mu_i = \mu_0$ is tested against the alternative $\mathrm{A}_i : \mu_i \neq \mu_0$ (two-tailed). Tamhane-Dunnett's test statistics are given by

$$t_i \frac{\bar{X}_i - \bar{X}_0}{\left(s_0^2/n_0 + s_i^2/n_i\right)^{1/2}} ~~ (1 \leq i \leq k)$$

The null hypothesis is rejected if $|t_i| > T_{k v_i \rho_{ij} \alpha}$ (two-tailed), with

$$v_i = n_0 + n_i - 2$$

degree of freedom and the correlation

$$\rho_{ii} = 1, ~ \rho_{ij} = 0 ~ (i \neq j).$$

The p-values are computed from the multivariate-t distribution as implemented in the function pmvt distribution.

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 `NA`s. Defaults to `getOption("na.action")`.

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

See Also

pmvt, welchManyOneTTest

Examples

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

fit <- aov(x ~ g - 1)
shapiro.test(residuals(fit))
bartlett.test(x ~ g - 1)
anova(fit)
## 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 `NA`s. Defaults to `getOption("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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j ~~ (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Tamhane T2 all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{\left(s^2_j/n_j + s^2_i/n_i\right)^{1/2}}, ~~ (i \ne j)$$

with s^2_i the variance of the i-th group. The null hypothesis is rejected (two-tailed) if

$$\Pr\left\{|t_{ij}| \ge t_{v_{ij}\alpha'/2}|\mathrm{H}\right\}_{ij} = \alpha.$$

T2 test uses Welch's approximate solution for calculating the degree of freedom.

$$v_{ij} = \frac{\left(s^2_i/n_i + s^2_j/n_j\right)^2}{s^4_i/n^2_i \left(n_i - 1\right) + s^4_j/n^2_j \left(n_j - 1\right)}.$$

T2' test applies the following approximation for the degree of freedom

$$v_{ij} = n_i + n_j - 2$$

The p-values are computed from the `TDist`-distribution and adjusted according to Dunn-Sidak.

$$p'_{ij} = \min\left\{1, ~ (1 - (1 - p_{ij})^m)\right\}$$

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.mehod = FALSE)`.

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.

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

References

Tamhane, A. C. (1979) A Comparison of Procedures for Multiple Comparisons of Means with Unequal Variances, *Journal of the American Statistical Association* **74**, 471–480.

See Also

`dunnettT3Test` `uryWigginsHochbergTest`

Examples

```
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
WT <- 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))
```

toTidy *Convert a PMCMR or osrt Object to a Data.Frame*

Description

The functions converts a list object of class "PMCMR" or "osrt" into a data.frame.

Usage

```
toTidy(mod, ...)
```

Arguments

mod	an object of class "PMCMR", "trendPMCMR" or "osrt".
...	further arguments. Currently ignored.

Value

A data.frame.

Author(s)

Indrajeet Patil (via email, 2020-1022), modified by Thorsten Pohlert

Examples

```
res <- tukeyTest(weight ~ Diet, data = ChickWeight, subset = Time == 21)
toTidy(res)
```

trout *Data from a Dose-Response Experiment with Trouts*

Description

This data set contains results from a dose-response experiment with trouts. The experiment was conducted with five doses of 10, 25, 60, 150 and 1000 ppm, respectively, plus a zero-dose control. The response is trout weight in mg.

Format

A data frame with 65 observations on the following 5 variables.

CONC a numeric vector of dose concentration in ppm

DOSE a factor with levels `1` `2` `3` `4` `5` `6`

REPA a factor with levels `1` `2`

REPC a factor with levels `1` `2`

Y a numeric vector of trout weight in mg

Source

ENV/JM/MONO(2006)18/ANN, page 113.

References

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

`tukeyTest` *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 `NA`s. Defaults to `getOption("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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j \quad (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Tukey's all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{s_{\mathrm{in}} \left(1/n_j + 1/n_i\right)^{1/2}}, \quad (i \ne j)$$

with s_{in}^2 the within-group ANOVA variance. The null hypothesis is rejected if $|t_{ij}| > q_{vm\alpha}/\sqrt{2}$, with $v = N - k$ degree of freedom. 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

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

Tukey, J. (1949) Comparing Individual Means in the Analysis of Variance, *Biometrics* **5**, 99–114.

See Also

`Tukey`, `TukeyHSD`

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 <- tukeyTest(fit)
summary(res)
summaryGroup(res)
```

`uryWigginsHochbergTest` *Ury, Wiggins, Hochberg Test*

Description

Performs Ury-Wiggins and Hochberg's all-pairs comparison test for normally distributed data with unequal variances.

Usage

```
uryWigginsHochbergTest(x, ...)

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

## S3 method for class 'formula'
uryWigginsHochbergTest(
  formula,
  data,
  subset,
  na.action,
  p.adjust.method = p.adjust.methods,
  ...
)

## 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 `NA`s. 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. Let X_{ij} denote a continuous random variable with the j-the realization ($1 \le j \le n_i$) in the i-th group ($1 \le i \le k$). Furthermore, the total sample size is $N = \sum_{i=1}^k n_i$. A total of $m = k(k-1)/2$ hypotheses can be tested: The null hypothesis is $\mathrm{H}_{ij} : \mu_i = \mu_j \quad (i \ne j)$ is tested against the alternative $\mathrm{A}_{ij} : \mu_i \ne \mu_j$ (two-tailed). Ury-Wiggins and Hochberg all-pairs test statistics are given by

$$t_{ij} \frac{\bar{X}_i - \bar{X}_j}{\left(s_j^2/n_j + s_i^2/n_i\right)^{1/2}}, \quad (i \ne j)$$

with s_i^2 the variance of the i-th group. The null hypothesis is rejected (two-tailed) if

$$\Pr\left\{|t_{ij}| \ge t_{v_{ij}\alpha'/2} |\mathrm{H}\right\}_{ij} = \alpha,$$

with Welch's approximate equation for degree of freedom as

$$v_{ij} = \frac{\left(s_i^2/n_i + s_j^2/n_j\right)^2}{s_i^4/n_i^2\left(n_i - 1\right) + s_j^4/n_j^2\left(n_j - 1\right)}.$$

The p-values are computed from the `TDist`-distribution. The type of test depends on the selected p-value adjustment method (see also `p.adjust`):

bonferroni the Ury-Wiggins test is performed with Bonferroni adjusted p-values.

hochberg the Hochberg test is performed with Hochberg's adjusted 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

Hochberg, Y. (1976) A Modification of the T-Method of Multiple Comparisons for a One-Way Layout With Unequal Variances, *Journal of the American Statistical Association* **71**, 200–203.

Ury, H. and Wiggins, A. D. (1971) Large Sample and Other Multiple Comparisons Among Means, *British Journal of Mathematical and Statistical Psychology* **24**, 174–194.

See Also

dunnettT3Test tamhaneT2Test TDist

Examples

```
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 `NA`s. Defaults to `getOption("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 $\mathrm{H}_{ij}: F_i(x) = F_j(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_{ij}: F_i(x) \ne F_j(x), ~~ i \ne j$. For `p.adjust.method = "single-step"` the Tukey's studentized range distribution is used to calculate p-values (see `Tukey`). Otherwise, the t-distribution is used for the calculation of p-values with a latter p-value adjustment as performed by `p.adjust`.

Value

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

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

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

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

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

alternative a character string describing the alternative hypothesis.

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

model a data frame of the input data.

dist a string that denotes the test distribution.

References

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

van der Waerden, B. L. (1952) Order tests for the two-sample problem and their power, *Indagationes Mathematicae* **14**, 453–458.

See Also

`vanWaerdenTest`, `vanWaerdenManyOneTest`, `normOrder`.

`vanWaerdenManyOneTest` *van-der-Waerden's Many-One Comparisons Normal Scores Test*

Description

Performs van-der-Waerden's multiple comparison normal scores test with one control.

Usage

```
vanWaerdenManyOneTest(x, ...)

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

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

Arguments

`x` a numeric vector of data values, or a list of numeric data vectors.

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

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

`alternative` the alternative hypothesis. Defaults to `two.sided`.

`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 `NA`s. Defaults to `getOption("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 $\mathrm{H}_i : F_0(x) = F_i(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_i : F_0(x) \ne F_i(x), \;\; 1 \le i \le k-1$. 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

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

van der Waerden, B. L. (1952) Order tests for the two-sample problem and their power, *Indagationes Mathematicae* **14**, 453–458.

See Also

vanWaerdenTest, vanWaerdenAllPairsTest, pmvt.

Examples

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

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

---

vanWaerdenTest *van der Waerden's Normal Scores Test*

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 to getOption("na.action").

Details

For one-factorial designs with non-normally distributed residuals van der Waerden's normal scores test can be performed to test the $\mathrm{H}_0 : F_1(x) = F_2(x) = \ldots = F_k(x)$ against the $\mathrm{H}_\mathrm{A} : F_i(x) \neq F_j(x)~(i \neq j)$ with at least one strict inequality.

Note

A tie correction is not applied in this function.

References

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

van der Waerden, B. L. (1952) Order tests for the two-sample problem and their power, *Indagationes Mathematicae* **14**, 453–458.

See Also

kruskalTest, normalScoresTest

Examples

```
vanWaerdenTest(count ~ spray, data = InsectSprays)
```

---

welchManyOneTTest *Welchs's Many-To-One Comparison Test*

Description

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

Usage

```
welchManyOneTTest(x, ...)

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

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

## S3 method for class 'aov'
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 `NA`s. 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 $\mathrm{H}_i : \mu_0(x) = \mu_i(x)$ is tested in the two-tailed test against the alternative $\mathrm{A}_i : \mu_0(x) \ne \mu_i(x), \;\; 1 \le i \le k - 1$.

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 adusted 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.

Welch, B. L. (1951) On the comparison of several mean values: An alternative approach, *Biometrika* **38**, 330–336.

See Also

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

Examples

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

fit <- aov(x ~ g)
shapiro.test(residuals(fit))
bartlett.test(x ~ g)
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

```
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"),
  ...
)

## S3 method for class 'aov'
williamsTest(x, 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 `NA`s. 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 $1 \le i \le m$, then the following $m = k - 1$ hypotheses are tested:

$$\begin{array}{ll}
\mathrm{H}_m : \bar{x}_0 = m_1 = \ldots = m_m, & \mathrm{A}_m : \bar{x}_0 \le m_1 \le \ldots m_m, \bar{x}_0 < m_m \\
\mathrm{H}_{m-1} : \bar{x}_0 = m_1 = \ldots = m_{m-1}, & \mathrm{A}_{m-1} : \bar{x}_0 \le m_1 \le \ldots m_{m-1}, \bar{x}_0 < m_{m-1} \\
\vdots & \vdots \\
\mathrm{H}_1 : \bar{x}_0 = m_1, & \mathrm{A}_1 : \bar{x}_0 < m_1,
\end{array}$$

where m_i denotes the isotonic mean of the ith dose level group.

William's test bases on a order restriction:

$$\mu_i^* = \max_{1 \le u \le i} \min_{i \le v \le m} \sum_{j=u}^{v} n_j \bar{x}_j^* \,/\, \sum_{j=u}^{v} n_j \qquad (1 \le i \le m),$$

where $\bar{x}_j^*$ denotes the j-th isotonic mean estimated with isotonic regression using the pool adjacent violators algorithm (PAVA) with the vector of means $\{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m\}^T$ and the vector of weights $\{n_1, n_2, \ldots, n_m\}^T$.

For the alternative hypothesis of decreasing trend, max and min are interchanged in the above Equation.

The i-the test statistic is calculated as follows:

$$\bar{t}_i = \frac{\mu_m^* - \bar{x}_0}{s_{\mathrm{E}} \sqrt{1/n_m - 1/n_0}}$$

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 β.

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 `"osrt"` that contains 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 statistic(s)

crit.value critical values for $\alpha = 0.05$.

alternative a character string describing the alternative hypothesis.

parameter the parameter(s) of the test distribution.

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**:

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

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

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

The Algorith AS 206 is available from StatLib https://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 excdeeds the number of replicates in the treatment levels. A warning message appears, if the following condition is not met, $1 \le n_0/n_i \le 6$ for $1 \le i \le m$.

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.osrt, summary.osrt

Examples

```
## Example from Sachs (1997, p. 402)
x <- c(106, 114, 116, 127, 145,
110, 125, 143, 148, 151,
136, 139, 149, 160, 174)
g <- gl(3,5)
levels(g) <- c("0", "I", "II")

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

Index

∗ **datasets**
algae, 11
Pentosan, 154
qPCR, 170
reviewers, 173
trout, 206
∗ **distribution**
Cochran, 34
Dgrubbs, 39
Grubbs, 83
Mandel-h, 125
Mandel-k, 126
∗ **friedmanranks**
durbinAllPairsTest, 49
durbinTest, 51
frdAllPairsConoverTest, 55
frdAllPairsExactTest, 58
frdAllPairsMillerTest, 60
frdAllPairsNemenyiTest, 63
frdAllPairsSiegelTest, 65
frdHouseTest, 67
frdManyOneDemsarTest, 70
frdManyOneExactTest, 72
frdManyOneNemenyiTest, 74
∗ **htest**
adAllPairsTest, 5
adKSampleTest, 7
adManyOneTest, 9
bwsAllPairsTest, 13
bwsKSampleTest, 15
bwsManyOneTest, 18
bwsTrendTest, 20
chaAllPairsNashimotoTest, 23
chackoTest, 26
chenJanTest, 29
chenTest, 31
cochranTest, 35
cuzickTest, 37
doubleGrubbsTest, 40
dscfAllPairsTest, 42
duncanTest, 43
dunnettT3Test, 45
dunnettTest, 47
durbinAllPairsTest, 49
durbinTest, 51
flignerWolfeTest, 52
frdAllPairsConoverTest, 55
frdAllPairsExactTest, 58
frdAllPairsMillerTest, 60
frdAllPairsNemenyiTest, 63
frdAllPairsSiegelTest, 65
frdHouseTest, 67
frdManyOneDemsarTest, 70
frdManyOneExactTest, 72
frdManyOneNemenyiTest, 74
friedmanTest, 76
gamesHowellTest, 78
gesdTest, 80
goreTest, 81
grubbsTest, 84
GSTTest, 85
hartleyTest, 87
hayterStoneTest, 89
hsAllPairsTest, 92
johnsonTest, 95
jonckheereTest, 97
kruskalTest, 100
kwAllPairsConoverTest, 103
kwAllPairsDunnTest, 105
kwAllPairsNemenyiTest, 108
kwManyOneConoverTest, 110
kwManyOneDunnTest, 113
kwManyOneNdwTest, 116
leTest, 118
lsdTest, 121
mackWolfeTest, 123
mandelhTest, 127
mandelkTest, 128

manyOneUTest, 130
medianAllPairsTest, 132
mrrTest, 137
MTest, 139
normalScoresAllPairsTest, 141
normalScoresManyOneTest, 143
normalScoresTest, 145
NPMTest, 147
osrtTest, 150
pageTest, 153
power.dunnett.test, 156
power.tukey.test, 158
quadeAllPairsTest, 171
scheffeTest, 174
shanTest, 176
shirleyWilliamsTest, 178
siegelTukeyTest, 182
skillingsMackTest, 184
snkTest, 186
spearmanTest, 188
steelsKSampleTest, 191
steelTest, 193
stepDownTrendTest, 195
tamhaneDunnettTest, 201
tamhaneT2Test, 203
tukeyTest, 207
uryWigginsHochbergTest, 209
vanWaerdenAllPairsTest, 211
vanWaerdenManyOneTest, 213
vanWaerdenTest, 215
welchManyOneTTest, 216
williamsTest, 218
∗ **kruskalranks**
kruskalTest, 100
kwAllPairsConoverTest, 103
kwAllPairsDunnTest, 105
kwAllPairsNemenyiTest, 108
kwManyOneConoverTest, 110
kwManyOneDunnTest, 113
kwManyOneNdwTest, 116
mackWolfeTest, 123
∗ **methods**
summary.gesdTest, 198
summary.mandel, 198
summary.osrt, 199
summary.PMCMR, 199
summary.trendPMCMR, 200
summaryGroup, 200
∗ **misc**
powerMCTests, 161
powerOneWayTests, 162
∗ **nonparametric**
adAllPairsTest, 5
adKSampleTest, 7
adManyOneTest, 9
bwsAllPairsTest, 13
bwsKSampleTest, 15
bwsManyOneTest, 18
bwsTrendTest, 20
chaAllPairsNashimotoTest, 23
chackoTest, 26
chenJanTest, 29
chenTest, 31
cuzickTest, 37
dscfAllPairsTest, 42
durbinAllPairsTest, 49
durbinTest, 51
flignerWolfeTest, 52
frdAllPairsConoverTest, 55
frdAllPairsExactTest, 58
frdAllPairsMillerTest, 60
frdAllPairsNemenyiTest, 63
frdAllPairsSiegelTest, 65
frdHouseTest, 67
frdManyOneDemsarTest, 70
frdManyOneExactTest, 72
frdManyOneNemenyiTest, 74
friedmanTest, 76
goreTest, 81
GSTTest, 85
hayterStoneTest, 89
hsAllPairsTest, 92
johnsonTest, 95
jonckheereTest, 97
kwAllPairsConoverTest, 103
kwAllPairsDunnTest, 105
kwAllPairsNemenyiTest, 108
kwManyOneConoverTest, 110
kwManyOneDunnTest, 113
kwManyOneNdwTest, 116
leTest, 118
mackWolfeTest, 123
manyOneUTest, 130
medianAllPairsTest, 132
medianTest, 134
normalScoresAllPairsTest, 141

normalScoresManyOneTest, 143
normalScoresTest, 145
NPMTest, 147
pageTest, 153
quadeAllPairsTest, 171
shanTest, 176
shirleyWilliamsTest, 178
siegelTukeyTest, 182
skillingsMackTest, 184
spearmanTest, 188
steelsKSampleTest, 191
stepDownTrendTest, 195
vanWaerdenAllPairsTest, 211
vanWaerdenManyOneTest, 213
vanWaerdenTest, 215

∗ **normalscores**
normalScoresAllPairsTest, 141
normalScoresManyOneTest, 143
vanWaerdenAllPairsTest, 211
vanWaerdenManyOneTest, 213

∗ **outliers**
cochranTest, 35
doubleGrubbsTest, 40
gesdTest, 80
grubbsTest, 84

∗ **parametric**
duncanTest, 43
dunnettT3Test, 45
dunnettTest, 47
gamesHowellTest, 78
hartleyTest, 87
lsdTest, 121
MTest, 139
osrtTest, 150
scheffeTest, 174
snkTest, 186
tamhaneDunnettTest, 201
tamhaneT2Test, 203
tukeyTest, 207
uryWigginsHochbergTest, 209
welchManyOneTTest, 216

∗ **plot**
barPlot, 12
plot.PMCMR, 155

∗ **print**
print.gesdTest, 165
print.mandel, 165
print.osrt, 166
print.PMCMR, 167
print.powerOneWayPMCMR, 167
print.powerPMCMR, 168
print.trendPMCMR, 168

∗ **testpower**
powerMCTests, 161
powerOneWayTests, 162

∗ **trendtest**
bwsTrendTest, 20
chackoTest, 26
cuzickTest, 37
flignerWolfeTest, 52
johnsonTest, 95
jonckheereTest, 97
leTest, 118
shanTest, 176
shirleyWilliamsTest, 178
spearmanTest, 188
stepDownTrendTest, 195

∗ **umbrellatest**
mackWolfeTest, 123

∗ **univariate**
gesdTest, 80
grubbsTest, 84

∗ **utilities**
toTidy, 206

∗ **wilcoxonranks**
chenJanTest, 29
chenTest, 31
cuzickTest, 37
dscfAllPairsTest, 42
flignerWolfeTest, 52
manyOneUTest, 130

ad.pval, *6–8*, *11*
adAllPairsTest, 5, *8*, *11*
adKSampleTest, *6*, 7, *11*
adManyOneTest, *6*, *8*, 9
algae, 11
aov, *44*, *45*, *48*, *79*, *121*, *174*, *186*, *202*, *203*, *207*, *209*, *217*
approx, *151*, *180*, *220*, *221*

barPlot, 12
bartlett.test, *37*, *88*
boxplot, *156*
bws_cdf, *19*, *20*
bws_stat, *19*, *20*
bws_test, *14*

bwsAllPairsTest, 13, *16*, *22*
bwsKSampleTest, 15
bwsManyOneTest, *16*, 18, *22*
bwsTrendTest, 20

chaAllPairsNashimotoTest, 23
chackoTest, *23*, *25*, 26, *197*
chenJanTest, 29
chenTest, 31
chisq.test, *134–136*
Chisquare, *87*, *102*, *109*, *110*
Cochran, 34
cochranTest, 35
cuzickTest, 37, *197*

default *(*kwAllPairsDunnTest*)*, 105
Dgrubbs, 39
doubleGrubbsTest, 40
dscfAllPairsTest, 42
duncanTest, 43
Dunnett *(*qDunnett*)*, 169
dunnettT3Test, 45, *205*, *211*
dunnettTest, 47, *169*
durbinAllPairsTest, 49
durbinTest, *50*, 51, *82*, *185*

factor, *10*, *19*, *21*, *28*, *34*, *38*, *54*, *96*, *99*, *112*, *114*, *120*, *131*, *145*, *177*, *190*, *197*
FDist, *35*, *102*, *175*
fligner.test, *37*, *86*, *87*
flignerWolfeTest, 52, *192*, *195*
frdAllPairsConoverTest, 55, *59*, *61*, *64*, *66*
frdAllPairsExactTest, *57*, 58, *61*, *64*, *66*
frdAllPairsMillerTest, *57*, *59*, 60, *64*, *66*
frdAllPairsNemenyiTest, *57*, *59*, *61*, 62, *66*
frdAllPairsSiegelTest, *57*, *59*, *61*, *64*, 65
frdHouseTest, 67
frdManyOneDemsarTest, *69*, 70, *73*, *75*
frdManyOneExactTest, *69*, *71*, 72, *75*
frdManyOneNemenyiTest, *71*, *73*, 74
friedman.test, *57*, *59*, *61*, *64*, *66*, *69*, *71*, *73*, *75*, *77*
friedmanTest, *57*, *59*, *61*, *64*, *66*, *69*, *71*, *73*, *75*, 76, *82*, *154*, *172*, *185*

gamesHowellTest, 78
gesdTest, 80
goreTest, 81
Grubbs, *40*, 83
grubbsTest, 84
GSTTest, 85

hartleyTest, 87
hayterStoneTest, 89, *94*
hsAllPairsTest, *91*, 92

invisible, *165–168*

johnsonTest, 95, *197*
jonckheereTest, 97, *197*

kruskal.test, *22*, *28*, *39*, *54*, *96*, *99*, *102*, *120*, *178*, *190*
kruskalTest, *22*, *28*, *39*, *54*, *96*, *99*, 100, *105*, *107*, *109*, *110*, *112*, *115*, *117*, *120*, *146*, *163*, *178*, *190*, *216*
kwAllPairsConoverTest, 103, *107*, *110*
kwAllPairsDunnTest, *105*, 105, *110*
kwAllPairsNemenyiTest, *105*, *107*, 108
kwManyOneConoverTest, 110, *115*, *117*, *195*
kwManyOneDunnTest, *112*, 113, *117*, *195*
kwManyOneNdwTest, *112*, *115*, 116, *195*

leTest, 118, *197*
lsdTest, 121

mackWolfeTest, 123
Mandel-h, 125
Mandel-k, 126
mandelhTest, *126*, 127, *166*, *198*
mandelkTest, *127*, 128, *166*, *198*
manyOneUTest, 130, *195*
medianAllPairsTest, 132
medianTest, 134
model.frame, *5*, *7*, *10*, *13*, *15*, *19*, *21*, *24*, *26*, *32*, *36*, *38*, *42*, *44*, *46*, *48*, *53*, *63*, *79*, *86*, *87*, *90*, *93*, *96*, *98*, *101*, *104*, *106*, *108*, *111*, *114*, *116*, *119*, *121*, *123*, *127*, *129*, *130*, *133*, *135*, *137*, *140*, *142*, *144*, *146*, *148*, *151*, *174*, *176*, *179*, *183*, *187*, *188*, *191*, *194*, *196*, *202*, *204*, *208*, *210*, *212*, *214*, *215*, *217*, *219*
mrrTest, 137
MTest, 139, *149*, *152*
murakami_cdf, *19*, *20*
murakami_stat, *19*, *20*

Normal, *25*, *31*, *34*, *54*, *68*, *107*, *131*, *180*, *183*

normalScoresAllPairsTest, 141, *145*
normalScoresManyOneTest, *143*, 143
normalScoresTest, *143*, *145*, 145, *216*
normOrder, *143*, *145*, *213*
NPMTest, *25*, *141*, 147

optimise, *160*
order, *10*, *19*, *21*, *28*, *34*, *38*, *54*, *96*, *99*, *112*, *114*, *120*, *131*, *145*, *177*, *190*, *197*
osrtTest, *91*, *141*, 150

p.adjust, *5*, *6*, *10*, *13*, *14*, *18*, *19*, *29*, *30*, *32*, *33*, *50*, *56*, *58*, *65*, *66*, *70*, *72*, *104–107*, *110*, *111*, *114*, *116*, *117*, *130*, *131*, *133*, *142*, *144*, *161*, *171*, *210*, *212*, *214*, *217*, *218*
p.adjust.methods, *24*, *50*, *133*, *172*, *217*
pageTest, 153
pairwise.t.test, *122*, *218*
pairwise.wilcox.test, *43*, *195*
pcochran, *36*
pcochran (Cochran), 34
pdgrubbs, *41*
pdgrubbs (Dgrubbs), 39
pDunnett, *48*, *49*
pDunnett (qDunnett), 169
Pentosan, 154
pgrubbs, *84*
pgrubbs (Grubbs), 83
pHayStonLSA, *91*
pKruskalWallis, *86*, *87*, *102*
plot.mandel, 155
plot.PMCMR, 155
pmandelh, *127*, *128*
pmandelh (Mandel-h), 125
pmandelk, *129*
pmandelk (Mandel-k), 126
pmaxFratio, *88*
pmvnorm, *31*, *74*, *75*, *115*, *131*
pmvt, *46–49*, *112*, *117*, *144*, *145*, *169*, *201*, *203*, *214*, *215*
pnorm, *124*
pNormScore, *146*
power.anova.test, *164*
power.dunnett.test, 156
power.tukey.test, 158
power.williams.test, 159
powerMCTests, *157*, *158*, 161, *164*, *168*
powerOneWayTests, 162, *168*
print, *165–168*
print (print.mandel), 165
print.gesdTest, 165
print.mandel, 165
print.osrt, 166, *195*, *199*, *221*
print.PMCMR, 167, *199*
print.powerOneWayPMCMR, 167
print.powerPMCMR, 168
print.trendPMCMR, 168, *200*
pwr.anova.test, *164*

qcochran (Cochran), 34
qDunnett, 169
qgrubbs (Grubbs), 83
qmandelh, *127*, *128*
qmandelh (Mandel-h), 125
qmandelk, *129*
qmandelk (Mandel-k), 126
qmvt, *157*, *169*
qPCR, 170
quade.test, *172*
quadeAllPairsTest, 171

reviewers, 173

sample, *16*, *21*, *22*, *91*, *94*, *124*
scheffeTest, 174
set.seed, *75*
shanTest, 176
shirleyWilliamsTest, *22*, *28*, *39*, *54*, *96*, *99*, *120*, *178*, 178, *190*, *195*
siegelTukeyTest, 182
skillingsMackTest, *82*, 184
snkTest, 186
spearmanTest, 188, *197*
Steel.test, *192*
steelsKSampleTest, 191
steelTest, 193
stepDownTrendTest, 195
summary.gesdTest, 198
summary.mandel, 198
summary.osrt, *195*, 199, *221*
summary.PMCMR, 199, *201*
summary.trendPMCMR, 200
summaryGroup, *199*, 200

t.test, *217*, *218*
tamhaneDunnettTest, 201, *218*
tamhaneT2Test, 203, *211*

TDist, *83*, *105*, *112*, *115*, *117*, *122*, *157*, *158*, *204*, *210*, *211*, *220*, *221*
toTidy, 206
trout, 206
Tukey, *43–45*, *79*, *80*, *105*, *109*, *110*, *142*, *158*, *187*, *208*, *212*
TukeyHSD, *45*, *187*, *208*
tukeyTest, *45*, *175*, *187*, 207

uryWigginsHochbergTest, *205*, 209

vanWaerdenAllPairsTest, 211, *215*
vanWaerdenManyOneTest, *213*, 213
vanWaerdenTest, *146*, *213*, *215*, 215
vanWeardenTest.default
(vanWaerdenTest), 215

welchManyOneTTest, *203*, 216
wilcox.test, *34*, *131*, *195*
Wilcoxon, *54*, *183*
williamsTest, *160*, *181*, 218