Package ‘nparcomp’

March 12, 2015

Type Package
Title Multiple Comparisons and Simultaneous Confidence Intervals
Version 2.6
Date 2015-03-12
Author Frank Konietschke
Maintainer Frank Konietschke <fkoniet@gwdg.de>
Description With this package, it is possible to compute nonparametric simultaneous confidence intervals for relative contrast effects in the unbalanced one way layout. Moreover, it computes simultaneous p-values. The simultaneous confidence intervals can be computed using multivariate normal distribution, multivariate t-distribution with a Satterthwaite Approximation of the degree of freedom or using multivariate range preserving transformations with Logit or Probit as transformation function. 2 sample comparisons can be performed with the same methods described above. There is no assumption on the underlying distribution function, only that the data have to be at least ordinal numbers.

License GPL
LazyLoad yes
Depends multcomp, mvtnorm
NeedsCompilation no
Repository CRAN
Date/Publication 2015-03-12 17:09:01

R topics documented:

  nparcomp-package ........................................... 2
  appetite ....................................................... 4
  colu .......................................................... 5
  gao ............................................................ 5
  gao_cs ........................................................ 7
  impla .......................................................... 8
  liver .......................................................... 9
  mctp .......................................................... 10
  mctp.rm ....................................................... 13
Description

With this package, it is possible to compute nonparametric simultaneous confidence intervals for relative contrast effects in the unbalanced one way layout. Moreover, it computes simultaneous p-values. The simultaneous confidence intervals can be computed using multivariate normal distribution, multivariate t-distribution with a Satterthwaite Approximation of the degree of freedom or using multivariate range preserving transformations with Logit or Probit as transformation function. 2 sample comparisons can be performed with the same methods described above. There is no assumption on the underlying distribution function, only that the data have to be at least ordinal numbers.

Details

Package:  nparcomp
Type:  Package
Version:  1.0-0
Date:  2012-06-22
License:  GPL

Author(s)

Frank Konietschke
Maintainer: Frank Konietschke <fkoniet@gwdg.de>

References


Examples

```r
# two sample comparisons: Nonparametric Behrens-Fisher Problem

data(impla)
a <- npar.t.test(impla~group, data = impla, method = "t.app",
                   alternative = "two.sided")
summary(a)
plot(a)

#--Analysis of relative contrast effects in different contrast settings

data(liver)

# Williams Contrast

a <- nparcomp(weight ~dosage, data=liver, asy.method = "probit",
              type = "Williams", alternative = "two.sided",
              plot.simci = TRUE, info = FALSE)
summary(a)

# Dunnett dose 3 is baseline

c <- nparcomp(weight ~dosage, data=liver, asy.method = "probit",
              type = "Dunnett", control = "3", alternative = "two.sided",
              plot.simci = TRUE, info = FALSE)
summary(c)

data(colu)

# Tukey comparison - one sided(lower)

a <- nparcomp(corpora~dose, data=colu, asy.method = "mult.t",
              type = "Tukey",alternative = "less")
```
appetite

Appetite scores of colorectal cancer patients

Description

Data from one of the quality of life measurements collected from colorectal cancer patients enrolled in the North Central Cancer Treatment Group phase III trials N9741. The patient received three treatment regimens: IFL (irinotecan, bolus fluorouracil, and leucovorin), FOLFOX (infused fluorouracil, leucovorin, and oxaliplatin), and IROX (irinotecan and oxaliplatin).

Usage

data(appetite)

Format

A data frame with 174 observations on the following 2 variables.

Group  A factor with levels FOLFOX IFL IROX.
Score   A numeric vector containing the appetite scores.

Details

The objective is to test whether there are differences between the treatment regimens in terms of different appetite scores.

Source


Examples

library(nparcomp)
data(appetite)
Numbers of corpora lutea

Description

Data from a fertility trial with 92 female Wistar rats: numbers of the corpora lutea in a placebo group and in 4 dose groups with an increasing dose of an active treatment.

Usage

data(colu)

Format

A data frame with 92 observations on the following 2 variables.

dose  A factor with levels dose1, dose2, dose3, dose4, placebo, where Placebo is the placebo group and dose1-dose4 are the 4 dose groups with an increasing dose.
corpora  A numeric vector containing the numbers of the corpora lutea.

Details

The objective is to test if the active treatment influences the fertility of the rats.

Source


Examples

library(nparcomp)
data(colu)
boxplot(corpora~dose,data=colu)

gao

Nonparametric multiple test procedure for many-to-one comparisons

Description

This function can be used to perform the nonparametric multiple tests for many-to-one comparisons by Gao et al. (2008). The multiple level is strongly controlled by the Hochberg-adjusment.

Usage

gao(formula, data, alpha = 0.05, control = NULL, silent = FALSE)
Arguments

formula A two-sided 'formula' specifying a numeric response variable and a factor with
more than two levels. If the factor contains less than 3 levels, an error message
will be returned.
data A dataframe containing the variables specified in formula.
alpha The significance level (by default = 0.05).
control Character string defining the control group in Dunnett comparisons. By default
it is the first group by lexicographical ordering
silent A logical indicating more informations should be print on screen.

Value

Info Samples and sizes with estimated relative effects and variance estimators.
Analysis Comparison: Distributions being compared, Estimator: Estimated effect, df:
Degree of Freedom, Statistic: Teststatistic, P.Raw: Raw p-Value P.Hochberg:
Adjusted p-Value by the Hochberg adjustment, Rejected: A logical indicating
rejected hypotheses, P.Bonf: Bonferroni adjusted p-Values, P.Holm: Holm ad-
justed p-Value.

Note

The procedure can only be used to test hypotheses in terms of the distribution functions.

Author(s)

Frank Konietschke

References

Gao, X. et al. (2008). Nonparametric Multiple Comparison Procedures for Unbalanced One-Way
Factorial Designs. JSPI 138, 2574 - 2591.
Package for Nonparametric Multiple Comparisons and Simultaneous Confidence Intervals. Journal
of Statistical Software, 61(10), 1-17.

See Also

For nonparametric all-pairs comparison see gao_cs.

Examples

data(liver)
gao(weight ~dosage, data=liver, alpha=0.05)

# Control= 3
gao(weight ~dosage, data=liver, alpha=0.05, control="3")
Nonparametric multiple test procedure for all-pairs comparisons

Description

This function can be used to perform the nonparametric multiple tests for all-pairs comparisons by Gao et al. (2008). This procedure is a nonparametric equivalent of Campbell and Skillings (1981) sequential test procedure.

Usage

gao_cs(formula, data, alpha = 0.05, silent = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>formula</td>
<td>A two-sided 'formula' specifying a numeric response variable and a factor with more than two levels. If the factor contains less than 3 levels, an error message will be returned.</td>
</tr>
<tr>
<td>data</td>
<td>A dataframe containing the variables specified in formula.</td>
</tr>
<tr>
<td>alpha</td>
<td>The significance level (by default = 0.05).</td>
</tr>
<tr>
<td>silent</td>
<td>A logical indicating more informations should be print on screen.</td>
</tr>
</tbody>
</table>

Value

Info: Samples and sizes with estimated relative effects and variance estimators.


Note

The generalized Campbell and Skillings’ analysis is performed in the CS.Analysis output. The adjusted quantiles and p-Values are reported. Due to the non-monotonicity of the adjusted quantiles, all results are checked for non-logical relations.

Author(s)

Frank Konietschke
References


See Also

For nonparametric many-to-one comparison see gao.

Examples

```r
data(reaction)
gao_cs(Time ~ Group, data=reaction, alpha=0.05)
```

Data frame:

<table>
<thead>
<tr>
<th>impla</th>
<th>Numbers of implantations</th>
</tr>
</thead>
</table>

Description

Data from a fertility trial with 29 female Wistar rats: numbers of the implantations in a placebo group and in an active treatment group.

Usage

```r
data(impla)
```

Format

A data frame with 29 observations on the following 2 variables.

group  A factor with levels Placebo, Verum, where Verum denotes the active treatment group.

impla  A numeric vector.

Details

The objective is to test if the active treatment influences the fertility of the rats.

Source

Examples

```r
library(nparcomp)
data(impla)
boxplot(impla~group,data=impla)
```

---

### Description

Data from a toxicity trial with male Wistar rats: Relative liver weights in a negative control group and in 4 dose groups with an increasing dose of an active treatment. After treatment the relative liver weights of the rats were computed.

### Usage

```r
data(liver)
```

### Format

A data frame with 38 observations on the following 2 variables.

- **dosage**: A numeric vector indicating the dose/control group.
- **weight**: A numeric vector containing the relative liver weights.

### Details

The objective is to test if the active treatment influences the liver weight of the rats.

### Source


### Examples

```r
data(liver)
boxplot(weight~dosage,data=liver)
```
mctp

Nonparam. multiple contrast tests and simult. confidence intervals

Description

The function mctp computes the estimator of nonparametric relative effects based on global rankings, simultaneous confidence intervals for the effects and adjusted p-values based on special contrasts like "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams", "UserDefined". The statistics are computed using multivariate normal distribution, multivariate Satterthwaite t-Approximation and multivariate transformations (Fisher function). The function 'mctp' also computes one-sided and two-sided confidence intervals and p-values. The confidence intervals can be plotted.

Usage

mctp(formula, data, type = c("Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams", "UserDefined"), conf.level = 0.95, alternative = c("two.sided", "less", "greater"), asy.method = c("fisher", "mult.t", "normal"), plot.simci = FALSE, control = NULL, info = TRUE, rounds = 3, contrast.matrix = NULL, correlation = FALSE, effect = c("unweighted", "weighted"))

Arguments

formula A two-sided 'formula' specifying a numeric response variable and a factor with more than two levels. If the factor contains less than 3 levels, an error message will be returned.
data A dataframe containing the variables specified in formula.
type Character string defining the type of contrast. It should be one of "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams", "UserDefined".
conf.level The confidence level for conf.level-confidence intervals (default is 0.95).
alternative Character string defining the alternative hypothesis, one of "two.sided", "less" or "greater".
asy.method Character string defining the asymptotic approximation method, one of "mult.t" for using a multivariate t-distribution with a Satterthwaite Approximation, "fisher" for using the Fisher transformation function, "normal", for using the multivariate normal distribution.
plot.simci A logical indicating whether you want a plot of the confidence intervals.
control Character string defining the control group in Dunnett comparisons. By default it is the first group by definition of the factor variable.
info A logical whether you want a brief overview with informations about the output.
rounds Number of rounds for the numeric values of the output (default is 3).

contrast.matrix User defined contrast matrix.

correlation A logical whether the estimated correlation matrix and covariance matrix should be printed.

effect Character string defining the type of effect, one of "unweighted" and "weighted".

Value

Data.Info List of samples and sample sizes and estimated effect per group.

Contrast Contrast matrix.


input List of input by user.

Note

If the samples are completely separated the variance estimators are Zero by construction. In these cases the Null-estimators are replaced by 0.001. Estimated relative effects with 0 or 1 are replaced with 0.001, 0.999 respectively.

A summary and a graph can be created separately by using the functions `summary.mctp` and `plot.mctp`.

For the analysis, the R packages `multcomp` and `mvtnorm` are required.

Author(s)

Frank Konietschke

References


See Also

For simultaneous confidence intervals for relative contrast effects, see `nparcomp`.
Examples

data(liver)

  # Williams Contrast
  a <- mctp(weight ~ dosage, data = liver, asy.method = "fisher",
            type = "Williams", alternative = "two.sided",
            plot.simci = TRUE, info = FALSE)
  summary(a)

  # Dunnett Contrast
  b <- mctp(weight ~ dosage, data = liver, asy.method = "fisher",
            type = "Dunnett", alternative = "two.sided",
            plot.simci = TRUE, info = FALSE)
  summary(b)

  # Dunnett dose 3 is baseline
  c <- mctp(weight ~ dosage, data = liver, asy.method = "fisher",
            type = "Dunnett", control = "3", alternative = "two.sided",
            plot.simci = TRUE, info = FALSE)
  summary(c)

data(colu)

  # Tukey comparison- one sided(less)
  a <- mctp(corpora~ dose, data = colu, asy.method = "mult.t",
            type = "Tukey", alternative = "less",
            plot.simci = TRUE, info = FALSE)
  summary(a)

  # Tukey comparison- one sided(greater)
  b <- mctp(corpora~ dose, data = colu, asy.method = "mult.t",
            type = "Tukey", alternative = "greater",
            plot.simci = TRUE, info = FALSE)
  summary(b)

  # Tukey comparison- one sided(less)
  c <- mctp(corpora~ dose, data = colu, asy.method = "mult.t",
            type = "Tukey", alternative = "less",
            plot.simci = TRUE, info = FALSE)
  summary(c)

  # Marcus comparison- one sided(greater)
  d <- mctp(corpora~ dose, data = colu, asy.method = "fisher",
**mctp.rm**

mctp.rm(formula, data, type = c("UserDefined", "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams"), control = NULL, conf.level = 0.95, alternative = c("two.sided", "lower", "greater"), rounds = 3, correlation = FALSE, asy.method = c("fisher", "normal", "mult.t"), plot.simci = FALSE, info = TRUE, contrast.matrix = NULL)

**Arguments**

- **formula**: A two-sided 'formula' specifying a numeric response variable and a repeated measures factor with more than two levels. If the factor contains less than 3 levels, an error message will be returned.
- **data**: A dataframe containing the variables specified in formula.
- **type**: Character string defining the type of contrast. It should be one of "UserDefined", "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams".
- **control**: If type=Dunnett, specification of the factor code which should serve as control (first level is default).
- **conf.level**: The confidence level for conflevel-confidence intervals (default is 0.95).
- **alternative**: Character string defining the alternative hypothesis, one of "two.sided", "less" or "greater".
- **rounds**: Number of rounds for the numeric values of the output (default is 3).
- **correlation**: A logical whether the estimated correlation matrix and covariance matrix should be printed.

**Description**

In the setting of a repeated measures design with n independent individuals and d repeated measures the function mctp.rm computes the estimator of nonparametric relative effects based on global rankings. Simultaneous confidence intervals for the effects and adjusted p-values based on special contrasts like "UserDefined", "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams" are provided. The statistics are computed using multivariate normal distribution, multivariate Satterthwaite t-Approximation and multivariate transformations (Fisher function). The function 'mctp.rm' also computes one-sided and two-sided confidence intervals and p-values. The confidence intervals can be plotted.
asy.method  Character string defining the asymptotic approximation method, one of "mult.t" for using a multivariate t-distribution with a Satterthwaite Approximation, "fisher" for using the Fisher transformation function, "normal", for using the multivariate normal distribution.

plot.simci  A logical indicating whether you want a plot of the confidence intervals.

info  A logical whether you want a brief overview with informations about the output.

contrast.matrix  User defined contrast matrix.

Value

Data.Info  List of samples and sample sizes and estimated effect per repeated measures level.

Contrast  Contrast matrix.


input  List of input by user.

Note

Estimated relative effects with 0 or 1 are replaced with 0.001 and 0.999.

A summary and a graph can be created separately by using the functions summary.mctp.rm and plot.mctp.rm.

For the analysis, the R packages 'multcomp' and 'mvtnorm' are required.

Author(s)

Marius Placzek

References


See Also

To analyse simple one-way layouts with independent samples use mctp.

Examples

data(panic)
a<-mctp.rm(cgi~weekL data=panicL type = "Dunnett", alternative = "two.sided", asy.method = "mult.t", plot.simci = FALSE, info = FALSE, contrast.matrix = NULL)
The nonparametric Behrens-Fisher problem

Description

The function npar.t.test performs two sample tests for the nonparametric Behrens-Fisher problem, that is testing the hypothesis

\[ H_0 : p = 1/2 \]

where \( p \) denotes the relative effect of 2 independent samples and computes confidence intervals for the relative effect \( p \). The statistics are computed using standard normal distribution, Satterthwaite t-Approximation and variance stabilising transformations (Probit and Logit transformation function). For small samples there is also a studentized permutation test implemented. npar.t.test also computes one-sided and two-sided confidence intervals and p-values. The confidence interval can be plotted.

Usage

```r
npar.t.test(formula, data, conf.level = 0.95, alternative = c("two.sided", "less", "greater"){3}, rounds = 3, method = c("logit", "probit", "normal", "t.app", "permu"), plot.simci = FALSE, info = TRUE, nperm=10000)
```

Arguments

- `formula`: A two-sided 'formula' specifying a numeric response variable and a factor with two levels. If the factor contains more than two levels, an error message will be returned.
- `data`: A dataframe containing the variables specified in formula.
- `conf.level`: The confidence level (default is 0.95).
- `alternative`: Character string defining the alternative hypothesis, one of "two.sided", "less" or "greater".
- `rounds`: Number of rounds for the numeric values of the output (default is 3).
- `method`: Character string defining the (asymptotic approximation) method, one of "logit", for using the logit transformation function, "probit", for using the probit transformation function, "normal", for using the standard normal distribution or "t.app" for using a t-Distribution with a Satterthwaite Approximation. The studentized permutation test can be obtained by choosing "permu".
- `plot.simci`: A logical indicating whether you want a plot of the confidence interval.
info

A logical whether you want a brief overview with informations about the output.

nperm

The number of permutations for the studentized permutation test. By default it is nperm=10,000.

Value

Info

List of samples and sample sizes.

Analysis

Effect: relative effect p(a,b) of the two samples 'a' and 'b', Estimator: estimated relative effect, Lower: Lower limit of the confidence interval, Upper: Upper limit of the confidence interval, T: teststatistic p.Value: p-value for the hypothesis by the chosen approximation method.

input

List of input by user.

Note

If the samples are completely seperated the variance estimators are Zero by construction. In these cases the Null-estimators are replaced by a replacing method as proposed in the paper from Neubert and Brunner (2006). Estimated relative effects with 0 or 1 are replaced with 0.001, 0.999 respectively.

A summary and a graph can be created separately by using the functions summary.nparttest and plot.nparttest.

Author(s)

Frank Konietschke.

References


See Also

For multiple comparison procedures based on relative effects, see nparcomp.

Examples

data(impla)
a <- npar.t.test(impla ~ group, data = impla, method = "t.app",
               alternative = "two.sided", info=FALSE)
summary(a)
plot(a)
npar.t.test.paired

b <- npar.t.test(impla ~ group, data = impla, method = "permu",
               alternative = "two.sided", info = FALSE)
summary(b)
plot(b)

npar.t.test.paired  A 2-sample nonparametric studentized permutation test for paired data

Description

The function npar.t.test.paired performs a two sample studentized permutation test for paired data, that is testing the hypothesis

$$H_0 : p = 1/2$$

where p denotes the relative effect of 2 dependent samples, and computes a confidence interval for the relative effect p. In addition the Brunner-Munzel-Test accompanied by a confidence interval for the relative effect is implemented. npar.t.test.paired also computes one-sided and two-sided confidence intervals and p-values. The confidence interval can be plotted.

Usage

npar.t.test.paired(formula, data, conf.level = 0.95, alternative = c("two.sided", "less", "greater"), nperm=10000, rounds = 3,
                   info = TRUE, plot.simci = TRUE)

Arguments

formula     A two-sided 'formula' specifying a numeric response variable and a factor with two levels. If the factor contains more than two levels, an error message will be returned.
data        A dataframe containing the variables specified in formula.
conf.level  The confidence level (default is 0.95).
alternative Character string defining the alternative hypothesis, one of "two.sided", "less" or "greater".
nperm       The number of permutations for the studentized permutation test. By default it is nperm=10,000.
rounds      Number of rounds for the numeric values of the output (default is 3).
info        A logical whether you want a brief overview with informations about the output.
plot.simci  A logical indicating whether you want a plot of the confidence interval.
Value

Info  List of samples and sample sizes.
Analysis  Effect: relative effect p(a,b) of the two samples 'a' and 'b', p.hat: estimated relative effect, Lower: Lower limit of the confidence interval, Upper: Upper limit of the confidence interval, T: studentized test statistic p.value: p-value for the hypothesis.
input  List of input by user.

Note

A summary and a graph can be created separately by using the functions `summary.nparttestpaired` and `plot.nparttestpaired`.

Make sure that your dataset is ordered by subjects before applying `npar.t.test.paired`.

Author(s)

Frank Konietschke

References


See Also

For multiple comparison procedures based on relative effects, see `nparcomp`.

Examples

data(PGI)
a<-npar.t.test.paired(PGIscore~timepoint, data = PGI, alternative = "two.sided", info=FALSE, plot.simci=FALSE)
summary(a)
plot(a)
nparcomp

Description

The function nparcomp computes the estimator of nonparametric relative contrast effects, simultaneous confidence intervals for the effects and simultaneous p-values based on special contrasts like "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams", "UserDefined". The statistics are computed using multivariate normal distribution, multivariate Satterthwaite t-Approximation and multivariate transformations (Probit and Logit transformation function). The function 'nparcomp' also computes one-sided and two-sided confidence intervals and p-values. The confidence intervals can be plotted.

Usage

nparcomp(formula, data, type = c("Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams", "UserDefined"), control = NULL, conf.level = 0.95, alternative = c("two.sided", "less", "greater"), rounds = 3, correlation = FALSE, asy.method = c("logit", "probit", "normal", "mult.t"), plot.simci = FALSE, info = TRUE, contrast.matrix=NULL, weight.matrix=FALSE)

Arguments

- **formula**: A two-sided 'formula' specifying a numeric response variable and a factor with more than two levels. If the factor contains less than 3 levels, an error message will be returned.
- **data**: A dataframe containing the variables specified in formula.
- **type**: Character string defining the type of contrast. It should be one of "Tukey", "Dunnett", "Sequen", "Williams", "Changepoint", "AVE", "McDermott", "Marcus", "UmbrellaWilliams", "UserDefined".
- **control**: Character string defining the control group in Dunnett comparisons. By default it is the first group by definition of the dataset.
- **conf.level**: The confidence level for the conflevel confidence intervals (default is 0.95).
- **alternative**: Character string defining the alternative hypothesis, one of "two.sided", "less" or "greater".
- **rounds**: Number of rounds for the numeric values of the output. By default it is rounds=3.
- **correlation**: A logical whether the estimated correlation matrix and covariance matrix should be printed.
- **asy.method**: Character string defining the asymptotic approximation method, one of "logit", for using the logit transformation function, "probit", for using the probit transformation function, "normal", for using the multivariate normal distribution or "mult.t" for using a multivariate t-distribution with a Satterthwaite Approximation.
- **plot.simci**: A logical indicating whether you want a plot of the confidence intervals.
- **info**: A logical whether you want a brief overview with informations about the output.
- **contrast.matrix**: User defined contrast matrix.
weight.matrix  A logical indicating whether the weight matrix should be printed.

Value

Data.Info  List of samples and sample sizes.
Contrast  Contrast matrix.
Analysis  

input  List of input by user.

Note
If the samples are completely separated the variance estimators are Zero by construction. In these cases the Null-estimators are replaced by 0.001. Estimated relative effects with 0 or 1 are replaced with 0.001, 0.999 respectively.

A summary and a graph can be created separately by using the functions summary.nparcomp and plot.nparcomp.

For the analysis, the R packages 'multcomp' and 'mvtnorm' are required.

Author(s)
Frank Konietschke

References

See Also
For two-sample comparisons based on relative effects, see npar.t.test.

Examples

data(liver)

  # Williams Contrast
  ac<-nparcomp(weight ~dosage, data=liver, asy.method = "probit",
          type = "Williams", alternative = "two.sided",
          plot.simci = TRUE, info = FALSE,correlation=TRUE)
  summary(ac)
# Dunnett dose 3 is baseline

c <- nparcomp(weight ~ dosage, data=liver, asy.method = "probit",
            type = "Dunnett", control = "3",
            alternative = "two.sided", info = FALSE)
summary(c)
plot(c)

data(colu)

# Tukey comparison- one sided(lower)

a <- nparcomp(corpora~ dose, data=colu, asy.method = "mult.t",
              type = "Tukey", alternative = "less",
              plot.simci = TRUE, info = FALSE)
summary(a)

# Tukey comparison- one sided(greater)

b <- nparcomp(corpora~ dose, data=colu, asy.method = "mult.t",
              type = "Tukey", alternative = "greater",
              plot.simci = TRUE, info = FALSE)
summary(b)

---

panic Clinical Global Impression (CGI) Scores

**Description**

Scores for the clinical global impression (CGI) measured on an ordinal scale (ranging from 2 to 8) during eight weeks for 16 patients with panic disorder attacks in a psychiatric clinical trial.

**Usage**

data(panic)

**Format**

A data frame with 80 observations on the following 2 variables.

- **CGI** A numeric vector containing the CGI score.
- **week** A numeric vector indicating the week (0,2,4,6,8) of measurement.
Details

Note that the first observation in each week corresponds to the first patient, the second one to the second patient, and so on. There are 5 repeated measures per patient.

Source


Examples

data(panic)
boxplot(CGI~week, data=panic)

---

Patient Rated Global Impression (PGI) Scores

Description

Scores for the patient rated global impression (PGI) measured on an ordinal scale (ranging from 1 to 6) being observed at baseline and after 4 weeks of treatment. The lower the score, the better the clinical impression.

Usage

data(PGI)

Format

A data frame with 30 observations on the following 3 variables.

- patient  A numeric vector indicating the patients.
- timepoint  A numeric vector indicating the week (0,2,4,6,8) of measurement.
- PGIscore  A numeric vector containing the PGI score.

Source


Examples

data(PGI)
boxplot(PGIscore~timepoint, data=PGI)
Description

This function takes an object of class "mctp" and creates a plot of the confidence intervals for the estimated effects.

Usage

```r
## S3 method for class 'mctp'
plot(x, ...)  
```

Arguments

- `x` An object of class "mctp", i.e. the result when applying `mctp` to a dataset. Otherwise an error will occur.
- `...` Arguments to be passed to methods.

Details

It is not possible to change any parameter set in the `mctp`-statement.

Since `plot.mctp` is a S3 method it suffices to use `plot(x)` as long as `x` is of class "mctp". It will be interpreted as `plot.mctp(x)`.

Value

`plot.mctp` returns a graph that contains a confidence interval for the estimated effect of each contrast. It just visualizes the result of the `mctp`-statement.

Note

It is possible to create a graphical result of the multiple comparison test procedure directly by setting `plot.simci=TRUE` in the `mctp`-statement.

To get a complete result summary of `mctp` the function `summary.mctp` can be used.

Author(s)

Frank Konietschke

References


See Also

For further information on the usage of `mctp`, see `mctp`. 
Examples

```r
data(liver)
a <- mctp(weight ~ dosage, data = liver, asy.method = "fisher",
          type = "Dunnett", alternative = "two.sided", plot.simci = FALSE,
          info = FALSE)
plot(a)
```

---

### plot.mctp.rm

**Visualizing the result of mctp.rm**

### Description

This function takes an object of class "mctp.rm" and creates a plot of the confidence intervals for the estimated effects.

### Usage

```r
## S3 method for class 'mctp.rm'
plot(x, ...)  
```

### Arguments

- `x`: An object of class "mctp.rm", i.e. the result when applying `mctp.rm` to a dataset.
  Otherwise an error will occur.
- `...`: Arguments to be passed to methods.

### Details

It is not possible to change any parameter set in the `mctp.rm`-statement.

Since `plot.mctp.rm` is a S3 method it suffices to use `plot(x)` as long as `x` is of class "mctp.rm". It will be interpreted as `plot.mctp.rm(x)`.

### Value

`plot.mctp.rm` returns a graph that contains a confidence interval for the estimated effect of each contrast. It just visualizes the result of the `mctp.rm`-statement.

### Note

It is possible to create a graphical result of the multiple comparison test procedure directly by setting `plot.simci=TRUE` in the `mctp.rm`-statement.

To get a complete result summary of `mctp.rm` the function `summary.mctp.rm` can be used.

### Author(s)

Marius Placzek
plot.nparcomp

References


See Also

For further information on the usage of mctp.rm, see mctp.rm.

Examples

data(panic)
a <- mctp.rm(CGI~week, data=panic, type = "Dunnett",
alternative = "two.sided",
asy.method = "fisher", contrast.matrix = NULL)
plot(a)

plot.nparcomp

Description

This function takes an object of class "nparcomp" and creates a plot of the confidence intervals for the estimated nonparametric contrast effects.

Usage

## S3 method for class 'nparcomp'
plot(x,...)

Arguments

x An object of class "nparcomp", i.e. the result when applying nparcomp to a dataset. Otherwise an error will occur.

... Arguments to be passed to methods.

Details

It is not possible to change any parameter set in the nparcomp-statement.

Since plot.nparcomp is a S3 method it suffices to use plot(x) as long as x is of class "nparcomp". It will be interpreted as plot.nparcomp(x).

Value

plot.nparcomp returns a graph that contains a confidence interval for the estimated nonparametric contrast effect of each contrast. It just visualizes the result of the nparcomp-statement.
Note

It is possible to create a graphical result directly by setting plot.simci=TRUE in the nparcomp-
statement.

Author(s)

Frank Konietschke

References


See Also

For further information on the usage of nparcomp, see nparcomp.

Examples

data(liver)
a<-nparcomp(weight ~dosageL data=liver, asy.method = "probit",
type = "Williams", alternative = "two.sided",
plot.simci = FALSE, info = FALSE)
plot(a)

plot.nparttest  Visualizing the result of npar.t.test

Description

This function takes an object of class "nparttest" and creates a plot of the confidence interval for the estimated effect.

Usage

## S3 method for class 'nparttest'
plot(x,...)

Arguments

x  x An object of class "nparttest", i.e. the result when applying npar.t.test to a dataset. Otherwise an error will occur.
...  ... Arguments to be passed to methods.
Details

It is not possible to change any parameter set in the `npar.t.test`-statement. Since `plot.nparttest` is a S3 method it suffices to use `plot(x)` as long as `x` is of class "nparttest". It will be interpreted as `plot.nparttest(x)`.

Value

`plot.npar.t.test` returns a graph that contains a confidence interval for the estimated effect of the nonparametric t-test. It just visualizes the result of the `npar.t.test`-statement.

Note

It is possible to create a graphical result of the nonparametric t-test directly by setting `plot.simci=TRUE` in the `npar.t.test`-statement.

Author(s)

Frank Konietschke

References


See Also

For further information on the usage of `npar.t.test`, see `npar.t.test`.

Examples

```r
data(impla)
a<-npar.t.test(impla~group, data = impla, method = "t.app",
              alternative = "two.sided", plot.simci=FALSE)
plot(a)
```

---

plot.nparttestpaired   Visualizing the result of `npar.t.test.paired`

Description

This function takes an object of class "nparttestpaired" and creates a plot of the confidence intervals for the estimated effect resulting from the studentized permutation test and the Brunner-Munzel test.
Usage

### S3 method for class 'npartestpaired'
plot(x,...)

Arguments

x x An object of class "npartestpaired", i.e. the result when applying `npar.t.test.paired` to a dataset. Otherwise an error will occur.

... Arguments to be passed to methods.

Details

It is not possible to change any parameter set in the `npar.t.test.paired`-statement.

Since plot.npartestpaired is a S3 method it suffices to use `plot(x)` as long as `x` is of class "npartestpaired". It will be interpreted as `plot.npartestpaired(x)`.

Value

`plot.npar.t.test` returns a graph that contains a confidence interval for the estimated effect of the nonparametric studentized permutation test as well as. It just visualizes the result of the `npar.t.test.paired`-statement.

Note

It is possible to create a graphical result of the nonparametric studentized permutation test directly by setting `plot.simci=TRUE` in the `npar.t.test.paired`-statement.

Author(s)

Frank Konietschke

References


See Also

For further information on the usage of `npar.t.test.paired`, see `npar.t.test.paired`.

Examples

data(PGI)
a<-npar.t.test.paired(PGIscore~timepoint, data = PGI,
alternative = "two.sided", info=TRUE, plot.simci=FALSE)
plot(a)
reaction

| reaction | Reaction times of mice [sec] |

Description

Data from a toxicity trial with 40 mice.

Usage

```r
data(reaction)
```

Format

A data frame with 40 observations on the following 2 variables.

- **Group**: A numeric vector indicating the group.
- **Time**: A numeric vector containing the reaction times.

Details

The objective is to test if the active treatment influences the reaction time of the mice.

Source


References


Examples

```r
library(nparcomp)
data(reaction)
boxplot(Time~Group,data=reaction)
```
Summary of mctp

Description
The function summary.mctp produces a result summary of mctp. It can only be applied to objects of class "mctp".

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

Arguments
- **object**: An object of class "mctp", i.e. the result when applying mctp to a dataset. Otherwise an error will occur.
- **...**: Arguments to be passed to methods.

Details
Since summary.mctp is a S3 method it suffices to use summary(x) as long as x is of class "mctp". It will be interpreted as summary.mctp(x).

Value
The function produces a summary of the result of mctp starting with some global information: alternative hypothesis, estimation method, type of contrast, confidence level. This is followed by:

- **Data.Info**: List of samples and sample sizes and estimated effect per group.
- **Contrast**: Contrast matrix.

Note
It is possible to create a graphical result of the multiple comparison test procedure by using the function plot.mctp.

Author(s)
Frank Konietschke
**References**


**See Also**

For further information on the usage of mctp, see `mctp`.

**Examples**

```r
data(liver)
a <- mctp(weight ~ dosage, data = liver, asy.method = "fisher",
          type = "Dunnett", alternative = "two.sided", plot.simci = FALSE,
          info = FALSE)
summary(a)
```

---

**Summary of mctp.rm**

**Description**

The function `summary.mctp.rm` produces a result summary of `mctp.rm`. It can only be applied to objects of class "mctp.rm".

**Usage**

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

**Arguments**

- `object` An object of class "mctp.rm", i.e. the result when applying `mctp.rm` to a dataset. Otherwise an error will occur.
- `...` Arguments to be passed to methods.

**Details**

Since `summary.mctp.rm` is a S3 method it suffices to use `summary(x)` as long as `x` is of class "mctp.rm". It will be interpreted as `summary.mctp.rm(x)`.

**Value**

The function produces a summary of the result of `mctp.rm` starting with some global information: alternative hypothesis, estimation method, type of contrast, confidence level. This is followed by:

- `Data.Info` List of samples and sample sizes and estimated effect per group.
- `Contrast` Contrast matrix.
Analysis


Note

It is possible to create a graphical result of the multiple comparison test procedure by using the function plot.mctp.rm.

Author(s)

Marius Placzek

References


See Also

For further information on the usage of mctp.rm, see mctp.rm.

Examples

data(panic)
a<-mctp.rm(CGI~week, data=panic, type = "Dunnett",
  alternative = "two.sided",
  asy.method = "fisher", contrast.matrix = NULL)
summary(a)

summary.nparcomp  Summary of nparcomp

Description

The function summary.nparcomp produces a result summary of nparcomp. It can only be applied to objects of class "nparcomp".

Usage

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

object An object of class "nparcomp", i.e. the result when applying nparcomp to a dataset. Otherwise an error will occur.

... Arguments to be passed to methods.

Details

Since summary.nparcomp is a S3 method it suffices to use summary(x) as long as x is of class "nparcomp". It will be interpreted as summary.nparcomp(x).

Value

The function produces a summary of the result of nparcomp starting with some global information: alternative hypothesis, estimation method, type of contrast, confidence level, method, interpretation. This is followed by:

Data.Info List of samples and sample sizes.
Contrast Contrast matrix.
Overall Overall p-value and critical value.

Note

It is possible to create a graphical result of the nonparametric test procedure nparcomp by using the function plot.nparcomp.

Author(s)

Frank Konietschke

References


See Also

For further information on the usage of nparcomp, see nparcomp.
### Examples

```r
data(liver)
a <- nparcomp(weight ~ dosage, data = liver, asy.method = "probit",
type = "Williams", alternative = "two.sided",
plot.simci = FALSE, info = FALSE)
summary(a)
```

### Description

The function `summary.npar.test` produces a result summary of `npar.test`. It can only be applied to objects of class "npar.test".

### Usage

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

### Arguments

- `object`: An object of class "npartest", i.e. the result when applying `npar.test` to a dataset. Otherwise an error will occur.
- `...`: Arguments to be passed to methods.

### Details

Since `summary.npar.test` is a S3 method it suffices to use `summary(x)` as long as `x` is of class "npartest". It will be interpreted as `summary.npar.test(x)`.

### Value

The function produces a summary of the result of `npar.test` starting with some global information: alternative hypothesis, confidence level, interpretation. This is followed by:

- **Info**: List of samples and sample sizes.
- **Analysis**: Effect: relative effect p(a,b) of the two samples 'a' and 'b', Estimator: estimated relative effect, Lower: Lower limit of the confidence interval, Upper: Upper limit of the confidence interval, T: teststatistic p.Value: p-value for the hypothesis by the choosen approximation method.
- **Permutation_Test**: Result of the studentized permutation test.

### Note

You can create a graphical result of the nonparametric t-test by using the function `plot.npar.test`.

---

**summary.npar.test**

Summary of **npar.test**

---

**Description**

The function `summary.npar.test` produces a result summary of `npar.test`. It can only be applied to objects of class "npar.test".

**Usage**

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

**Arguments**

- `object`: An object of class "npartest", i.e. the result when applying `npar.test` to a dataset. Otherwise an error will occur.
- `...`: Arguments to be passed to methods.

**Details**

Since `summary.npar.test` is a S3 method it suffices to use `summary(x)` as long as `x` is of class "npartest". It will be interpreted as `summary.npar.test(x)`.

**Value**

The function produces a summary of the result of `npar.test` starting with some global information: alternative hypothesis, confidence level, interpretation. This is followed by:

- **Info**: List of samples and sample sizes.
- **Analysis**: Effect: relative effect p(a,b) of the two samples 'a' and 'b', Estimator: estimated relative effect, Lower: Lower limit of the confidence interval, Upper: Upper limit of the confidence interval, T: teststatistic p.Value: p-value for the hypothesis by the choosen approximation method.
- **Permutation_Test**: Result of the studentized permutation test.

**Note**

You can create a graphical result of the nonparametric t-test by using the function `plot.npar.test`. 
**summary.nparttestpaired**

**Author(s)**
Frank Konietschke

**References**


**See Also**
For further information on the usage of npar.t.test, see `npar.t.test`.

**Examples**
```r
data(impla)
a<-npar.t.test(impla~group, data = impla, method = "t.app",
              alternative = "two.sided",
              plot.simci=FALSE, info=FALSE)
summary(a)
```

---

**Description**
The function `summary.nparttestpaired` produces a result summary of `npar.t.test.paired`. It can only be applied to objects of class "npartestpaired".

**Usage**
```r
## S3 method for class 'npartestpaired'
summary(object,...)
```

**Arguments**
- `object` An object of class "npartestpaired", i.e. the result when applying `npar.t.test.paired` to a dataset. Otherwise an error will occur.
- `...` Arguments to be passed to methods.

**Details**
Since `summary.nparttestpaired` is a S3 method it suffices to use `summary(x)` as long as x is of class "npartestpaired". It will be interpreted as `summary.nparttestpaired(x)`.
Value

The function produces a summary of the result of \texttt{npar.t.test.paired} starting with some global information: alternative hypothesis, confidence level, interpretation. This is followed by:

\textbf{Info} \quad List of samples and sample sizes.

\textbf{Analysis} \quad Effect: relative effect $p(a,b)$ of the two samples 'a' and 'b', $p.hat$: estimated relative effect, Lower: Lower limit of the confidence interval, Upper: Upper limit of the confidence interval, $T$: test statistic $p.value$: p-value for the hypothesis by the chosen approximation method.

Note

You can create a graphical result of the nonparametric paired t-test by using the function \texttt{plot.nparttestpaired}.

Author(s)

Frank Konietschke

References


See Also

For further information on the usage of \texttt{npar.t.test.paired}, see \texttt{npar.t.test.paired}.

Examples

data(PGI)
a<-npar.t.test.paired(PGIscore~timepoint, data = PGI,
                        alternative = "two.sided", info=FALSE, plot.simci=FALSE)
summary(a)
Index

*Topic aplot
  plot.mctp, 23
  plot.mctp.rm, 24
  plot.nparcomp, 25
  plot.nparttest, 26
  plot.nparttestpaired, 27
*Topic datasets
  appetite, 4
  colu, 5
  impla, 8
  liver, 9
  panic, 21
  PGI, 22
  reaction, 29
*Topic htest
  gao, 5
  gao_cs, 7
  mctp, 10
  mctp.rm, 13
  npar.t.test, 15
  npar.t.testpaired, 17
  nparcomp, 18
  nparcomp-package, 2
*Topic package
  nparcomp-package, 2
*Topic print
  summary.mctp, 30
  summary.mctp.rm, 31
  summary.nparcomp, 32
  summary.nparttest, 34
  summary.nparttestpaired, 35

appetite, 4

colu, 5

gao, 5, 8

gao_cs, 6, 7

impla, 8

liver, 9

mctp, 10, 14, 23, 30, 31
mctp.rm, 13, 24, 25, 31, 32

npar.t.test, 15, 20, 26, 27, 34, 35
npar.t.testpaired, 17, 27, 28, 35, 36
nparcomp, 11, 16, 18, 25, 26, 32, 33
nparcomp-package, 2

panic, 21

PGI, 22

plot.mctp, 11, 23, 30
plot.mctp.rm, 14, 24, 32
plot.nparcomp, 20, 25, 33
plot.nparttest, 16, 26, 34
plot.nparttestpaired, 18, 27, 36

reaction, 29

summary.mctp, 11, 23, 30
summary.mctp.rm, 14, 24, 31
summary.nparcomp, 20, 32
summary.nparttest, 16, 34
summary.nparttestpaired, 18, 35