Package ‘rcompanion’

July 27, 2019

Version 2.2.2
Date 2019-07-27
Title Functions to Support Extension Education Program Evaluation

URL http://rcompanion.org/
Depends R (>= 3.3.0)
Imports stats (>= 3.3.0), graphics (>= 3.3.0), boot (>= 1.3-18), DescTools (>= 0.99.17), multcompView (>= 0.1-7), plyr (>= 1.8.4), EMT (>= 1.1), coin (>= 1.1-2), lmtest (>= 0.9-34), nortest (>= 1.0-4)
Suggests FSA (>= 0.8.9)
Author Salvatore Mangiafico [aut, cre]
Maintainer Salvatore Mangiafico <mangiafico@njaes.rutgers.edu>
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 6.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2019-07-27 15:40:02 UTC

R topics documented:

accuracy .................. 2
Anderson .................. 3
AndersonBias ............... 4
AndersonRainBarrel .......... 4
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AndersonRainGarden</td>
<td>5</td>
</tr>
<tr>
<td>BobBelcher</td>
<td>5</td>
</tr>
<tr>
<td>Breakfast</td>
<td>6</td>
</tr>
<tr>
<td>BrendonSmall</td>
<td>6</td>
</tr>
<tr>
<td>BullyHill</td>
<td>7</td>
</tr>
<tr>
<td>Cathbus</td>
<td>7</td>
</tr>
<tr>
<td>cateNelson</td>
<td>8</td>
</tr>
<tr>
<td>clcList</td>
<td>10</td>
</tr>
<tr>
<td>cliffDelta</td>
<td>11</td>
</tr>
<tr>
<td>cohenG</td>
<td>13</td>
</tr>
<tr>
<td>cohenH</td>
<td>15</td>
</tr>
<tr>
<td>cohenW</td>
<td>16</td>
</tr>
<tr>
<td>compareGLM</td>
<td>17</td>
</tr>
<tr>
<td>compareLM</td>
<td>18</td>
</tr>
<tr>
<td>cramerV</td>
<td>20</td>
</tr>
<tr>
<td>cramerVFit</td>
<td>21</td>
</tr>
<tr>
<td>epsilonSquared</td>
<td>23</td>
</tr>
<tr>
<td>freemanTheta</td>
<td>25</td>
</tr>
<tr>
<td>fullPTable</td>
<td>27</td>
</tr>
<tr>
<td>groupwiseCMH</td>
<td>28</td>
</tr>
<tr>
<td>groupwiseGeometric</td>
<td>29</td>
</tr>
<tr>
<td>groupwiseHuber</td>
<td>31</td>
</tr>
<tr>
<td>groupwiseMean</td>
<td>32</td>
</tr>
<tr>
<td>groupwiseMedian</td>
<td>34</td>
</tr>
<tr>
<td>groupwisePercentile</td>
<td>36</td>
</tr>
<tr>
<td>groupwiseSum</td>
<td>38</td>
</tr>
<tr>
<td>HayleySmith</td>
<td>39</td>
</tr>
<tr>
<td>kendallW</td>
<td>40</td>
</tr>
<tr>
<td>Monarchs</td>
<td>41</td>
</tr>
<tr>
<td>multiVDA</td>
<td>42</td>
</tr>
<tr>
<td>nagelkerke</td>
<td>43</td>
</tr>
<tr>
<td>nagelkerkeHermite</td>
<td>45</td>
</tr>
<tr>
<td>nominalSymmetryTest</td>
<td>46</td>
</tr>
<tr>
<td>Nurseries</td>
<td>47</td>
</tr>
<tr>
<td>pairwiseDifferences</td>
<td>48</td>
</tr>
<tr>
<td>pairwiseMcnemar</td>
<td>48</td>
</tr>
<tr>
<td>pairwiseMedianMatrix</td>
<td>50</td>
</tr>
<tr>
<td>pairwiseMedianTest</td>
<td>51</td>
</tr>
<tr>
<td>pairwiseModelAnova</td>
<td>53</td>
</tr>
<tr>
<td>pairwiseNominalIndependence</td>
<td>54</td>
</tr>
<tr>
<td>pairwiseNominalMatrix</td>
<td>55</td>
</tr>
<tr>
<td>pairwiseOrdinalIndependence</td>
<td>56</td>
</tr>
<tr>
<td>pairwiseOrdinalMatrix</td>
<td>58</td>
</tr>
<tr>
<td>pairwiseOrdinalPairedMatrix</td>
<td>58</td>
</tr>
<tr>
<td>pairwiseOrdinalPairedTest</td>
<td>59</td>
</tr>
<tr>
<td>pairwiseOrdinalTest</td>
<td>59</td>
</tr>
<tr>
<td>pairwisePercentileTest</td>
<td>60</td>
</tr>
<tr>
<td>pairwisePermutationMatrix</td>
<td>62</td>
</tr>
</tbody>
</table>
accuracy

Minimum maximum accuracy, mean absolute percent error, root mean square error, coefficient of variation, and Efron’s pseudo r-squared

Description

Produces a table of fit statistics for multiple models.

Usage

accuracy(fits, plotit = TRUE, digits = 3, ...)

Arguments

fits A series of model object names. Must be a list.
plotit If TRUE, produces plots of the predicted values vs. the actual values for each model.
digits The number of significant digits in the output.
... Other arguments passed to plot.
Details

Produces a table of fit statistics for multiple models: minimum maximum accuracy, mean absolute percentage error, root mean square error, normalized root mean square error, accuracy based on normalized root mean square error, Efron’s pseudo r-squared, and coefficient of variation.

For minimum maximum accuracy, larger indicates a better fit, and a perfect fit is equal to 1.

For mean absolute error (MAE), smaller indicates a better fit, and a perfect fit is equal to 0. It has the same units as the dependent variable. Note that here, MAE is simply the mean of the absolute values of the differences of predicted values and the observed values ($\text{MAE} = \text{mean}(|\text{predy} - \text{actual}|)$). There are other definitions of MAE and similar-sounding terms.

For mean absolute percent error (MAPE), smaller indicates a better fit, and a perfect fit is equal to 0.

Root mean square error (RMSE) has the same units as the predicted values.

Normalized root mean square error (NRMSE) is RMSE divided by the mean or the median of the values of the dependent variable.

NRMSE accuracy values are calculated as 1 minus NRMSE. Larger indicates a better fit, and a perfect fit is equal to 1.

Efron’s pseudo r-squared is calculated as 1 minus the residual sum of squares divided by the total sum of squares. For linear models (lm model objects), Efron’s pseudo r-squared will be equal to r-squared. For other models, it should not be interpreted as r-squared, but can still be useful as a relative measure.

CV.percent is the coefficient of variation for the model. Here it is expressed as a percent.

Model objects currently supported: lm, glm, nls, betareg, gls, lme, lmer, lmerTest, rq, loess, gam, glm.nb, glmRob.

Value

A list of two objects: The series of model calls, and a data frame of statistics for each model.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References


See Also

compareLM, compareGLM, nagelkerke

Examples

data(BrendonSmall)
BrendonSmall$Calories = as.numeric(BrendonSmall$Calories)
BrendonSmall$Calories2 = BrendonSmall$Calories ^ 2
model.1 = lm(Sodium ~ Calories, data = BrendonSmall)
model.2 = lm(Sodium ~ Calories + Calories2, data = BrendonSmall)
model.3 = glm(Sodium ~ Calories, data = BrendonSmall, family="Gamma")
quadplat = function(x, a, b, clx) {
  ifelse(x < clx, a + b * x + (-0.5*b/clx) * x * x,
         a + b * clx + (-0.5*b/clx) * clx * clx)
}
model.4 = nls(Sodium ~ quadplat(Calories, a, b, clx),
              data = BrendonSmall,
              start = list(a=519, b=0.359, clx = 2300))
accuracy(list(model.1, model.2, model.3, model.4), plotit=FALSE)

### Perfect and poor model fits
X = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
Y = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
Z = c(1, 12, 13, 6, 10, 13, 4, 3, 5, 6, 10, 14)
perfect = lm(Y ~ X)
poor = lm(Z ~ X)
accuracy(list(perfect, poor), plotit=FALSE)

---

Anderson

**Hypothetical data for Alexander Anderson**

**Description**

A matrix of counts for students passing or failing a pesticide training course across four counties. Hypothetical data.

**Usage**

Anderson

**Format**

An object of class `matrix` with 4 rows and 2 columns.

**Source**

http://rcompanion.org/handbook/H_04.html
AndersonBias  
*Hypothetical data for Alexander Anderson with sex bias*

**Description**
A data frame of counts for students passing or failing a pesticide training course across four counties, with sex of students. Hypothetical data.

**Usage**
AndersonBias

**Format**
An object of class `data.frame` with 16 rows and 4 columns.

**Source**
http://rcompanion.org/handbook/H_06.html

AndersonRainBarrel  
*Hypothetical data for Alexander Anderson on rain barrel installation*

**Description**
A matrix of paired counts for students planning to install rain barrels before and after a class. Hypothetical data.

**Usage**
AndersonRainBarrel

**Format**
An object of class `matrix` with 2 rows and 2 columns.

**Source**
http://rcompanion.org/handbook/H_05.html
**AndersonRainGarden**  
*Hypothetical data for Alexander Anderson on rain garden installation*

### Description
A matrix of paired counts for students planning to install rain gardens before and after a class. Hypothetical data.

### Usage
AndersonRainGarden

### Format
An object of class *matrix* with 3 rows and 3 columns.

### Source
http://rcompanion.org/handbook/H_05.html

**BobBelcher**  
*Hypothetical data for ratings of instructors in unreplicated CBD*

### Description
A data frame of Likert responses for five instructors for each of 8 respondents. Arranged in unreplicated complete block design. Hypothetical data.

### Usage
BobBelcher

### Format
An object of class *data.frame* with 40 rows and 3 columns.

### Source
http://rcompanion.org/handbook/F_10.html
**Breakfast**

*Hypothetical data for students’ breakfast habits and travel to school*

**Description**

A two-dimensional contingency table, in which Breakfast is an ordered nominal variable, and Travel is a non-ordered nominal variable. Hypothetical data.

**Usage**

Breakfast

**Format**

An object of class `table` with 3 rows and 5 columns.

**Source**

http://rcompanion.org/handbook/H_09.html

---

**BrendonSmall**

*Hypothetical data for Brendon Small and company*

**Description**

A data frame of the intake of calories and sodium for students in five classes. Hypothetical data.

**Usage**

BrendonSmall

**Format**

An object of class `data.frame` with 45 rows and 6 columns.

**Source**

http://rcompanion.org/handbook/I_10.html
BullyHill

Hypothetical data for proportion of students passing a certification

Description
A data frame of counts of students passing and failing. Hypothetical data.

Usage
BullyHill

Format
An object of class `data.frame` with 12 rows and 5 columns.

Source
http://rcompanion.org/handbook/J_02.html

Catbus

Hypothetical data for Catbus and company

Description
A data frame of the number of steps taken by students in three classes. Hypothetical data.

Usage
Catbus

Format
An object of class `data.frame` with 26 rows and 5 columns.

Source
http://rcompanion.org/handbook/C_03.html
cateNelson  
---

**cateNelson**  
*Cate-Nelson models for bivariate data*

**Description**

Produces critical-x and critical-y values for bivariate data according to a Cate-Nelson analysis.

**Usage**

```r
cateNelson(x, y, plotit = TRUE, hollow = TRUE, xlab = "X",
ylab = "Y", trend = "positive", clx = 1, cly = 1,
xthreshold = 0.1, ythreshold = 0.1, progress = TRUE,
verbose = TRUE, listout = FALSE)
```

**Arguments**

- `x`: A vector of values for the x variable.
- `y`: A vector of values for the y variable.
- `plotit`: If TRUE, produces plots of the output.
- `hollow`: If TRUE, uses hollow circles on the plot to indicate data not fitting the model.
- `xlab`: The label for the x-axis.
- `ylab`: The label for the y-axis.
- `trend`: "positive" if the trend of y vs. x is generally positive. "negative" if negative.
- `clx`: Indicates which of the listed critical x values should be chosen as the critical x value for the final model.
- `cly`: Indicates which of the listed critical y values should be chosen as the critical y value for the final model.
- `xthreshold`: Indicates the proportion of potential critical x values to display in the output. A value of 1 would display all of them.
- `ythreshold`: Indicates the proportion of potential critical y values to display in the output. A value of 1 would display all of them.
- `progress`: If TRUE, prints an indicator of progress as for loops progress.
- `verbose`: If FALSE, suppresses printed output of tables.
- `listout`: If TRUE, outputs a list of data frames instead of a single data frame. This allows a data frame of critical values and associated statistics to be extracted, for example if one would want to sort by Cramer’s V.
Details

Cate-Nelson analysis divides bivariate data into two groups. For data with a positive trend, one group has a large \( x \) value associated with a large \( y \) value, and the other group has a small \( x \) value associated with a small \( y \) value. For a negative trend, a small \( x \) is associated with a large \( y \), and so on.

The analysis is useful for bivariate data which don’t conform well to linear, curvilinear, or plateau models.

This function will fail if either of the largest two or smallest two \( x \) values are identical.

Value

A data frame of statistics from the analysis: number of observations, critical level for \( x \), sum of squares, critical value for \( y \), the number of observations in each of the quadrants (I, II, III, IV), the number of observations that conform with the model, the proportion of observations that conform with the model, the number of observations that do not conform to the model, the proportion of observations that do not conform to the model, a p-value for the Fisher exact test for the data divided into the groups indicated by the model, and Cramer’s V for the data divided into the groups indicated by the model.

Output also includes printed lists of critical values, explanation of the values in the data frame, and plots: \( y \) vs. \( x \); sum of squares vs. critical \( x \) value; the number of observations that do not conform to the model vs. critical \( y \) value; and \( y \) vs. \( x \) with the critical values shown as lines on the plot, and the quadrants labeled.

Note


An earlier version of this function was published in Mangiafico, S.S. 2013. Cate-Nelson Analysis for Bivariate Data Using R-project. J.of Extension 51:5, STOT1.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/rcompanion/h_02.html

Examples

data(Nurseries)
cateNelson(x = Nurseries$Size, y = Nurseries$Proportion, plotit = TRUE, hollow = TRUE, xlab = "Nursery size in hectares", ylab = "Proportion of good practices adopted")
cldList

Compact letter display for lists of comparisons

Description

Produces a compact letter display (cld) from pairwise comparisons that were summarized in a table of comparisons.

Usage

```
cldList(formula = NULL, data = NULL, comparison = NULL, p.value = NULL, threshold = 0.05, print.comp = FALSE, remove.space = TRUE, remove.equal = TRUE, remove.zero = TRUE, swap.colon = TRUE, swap.vs = FALSE, ...)
```

Arguments

- `formula`: A formula indicating the variable holding p-values and the variable holding the comparisons. e.g. `P.adj ~ Comparison`.
- `data`: The data frame to use.
- `comparison`: A vector of text describing comparisons, with each element in a form similar to "Treat.A - Treat.B = 0". Spaces and "=" and "0" are removed by default.
- `p.value`: A vector of p-values corresponding to the comparisons in the `comparison` argument.
- `threshold`: The alpha value. That is, the p-value below which the comparison will be considered significant.
- `print.comp`: If TRUE, prints out a data frame of the modified text of the comparisons. Useful for debugging.
- `remove.space`: If TRUE, removes spaces from the text of the comparisons.
- `remove.equal`: If TRUE, removes "=" from the text of the comparisons.
- `remove.zero`: If TRUE, removes "0" from the text of the comparisons.
- `swap.colon`: If TRUE, swaps ".:" with "-" in the text of the comparisons.
- `swap.vs`: If TRUE, swaps "vs" with "-" in the text of the comparisons.
- `...`: Additional arguments passed to `multcompLetters`.

trend = "positive",
clx = 1,
xthreshold = 0.10,
ythreshold = 0.15)
cliffDelta

Details

The input should include either formula and data, or comparison and p.value.

This function relies upon the multcompLetters function in the multcompView package. The text for the comparisons passed to multcompLetters should be in the form "Treat.A-Treat.B". Currently cldList removes spaces, equal signs, and zeros, by default, and so can use text in the form e.g. "Treat.A - Treat.B = 0". It also changes ":" to ",", and so can use text in the form e.g. "Treat.A : Treat.B".

Value

A data frame of group names, group separation letters, and monospaced separations letters

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_08.html

Examples

data(PoohPiglet)
PoohPiglet$Speaker = factor(PoohPiglet$Speaker,
levels=c("Pooh", "Tigger", "Piglet"))

library(FSA)
DT = dunnTest(Likert ~ Speaker,
data=PoohPiglet,
method="bh")

DT = DT$res
DT
cldList(P.adj ~ Comparison,
data = DT,
threshold = 0.05)

cliffDelta Cliff’s delta

Description

Calculates Cliff’s delta with confidence intervals by bootstrap
Usage

cliffDelta(formula = NULL, data = NULL, x = NULL, y = NULL, 
  ci = FALSE, conf = 0.95, type = "perc", R = 1000, 
  histogram = FALSE, digits = 3, ...)

Arguments

  formula  A formula indicating the response variable and the independent variable. e.g. y ~ group.
  data     The data frame to use.
  x         If no formula is given, the response variable for one group.
  y         The response variable for the other group.
  ci        If TRUE, returns confidence intervals by bootstrap. May be slow.
  conf      The level for the confidence interval.
  type      The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.
  R         The number of replications to use for bootstrap.
  histogram If TRUE, produces a histogram of bootstrapped values.
  digits    The number of significant digits in the output.
  ...       Additional arguments passed to the wilcox.test function.

Details

Cliff’s delta is an effect size statistic appropriate in cases where a Wilcoxon-Mann-Whitney test might be used. It ranges from -1 to 1, with 0 indicating stochastic equality, and 1 indicating that the first group dominates the second. It is linearly related to Vargha and Delaney’s A.

The function calculates Cliff’s delta from the "W" U statistic from the wilcox.test function. Specifically, $VDA = U/(n1*n2)$; $CD = (VDA-0.5)*2$.

The input should include either formula and data; or x, and y. If there are more than two groups, only the first two groups are used.

Currently, the function makes no provisions for NA values in the data. It is recommended that NAs be removed beforehand.

When the data in the first group are greater than in the second group, Cliff’s delta is positive. When the data in the second group are greater than in the first group, Cliff’s delta is negative. Be cautious with this interpretation, as R will alphabetize groups in the formula interface if the grouping variable is not already a factor.

When Cliff’s delta is close to 1 or close to -1, or with small sample size, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, Cliff’s delta. Or a small data frame consisting of Cliff’s delta, and the lower and upper confidence limits.
Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_04.html

See Also

vda, multiVDA

Examples

data(Catbus)
cliffDelta(Steps ~ Sex, data=Catbus)

---

```r
cohenG(x, ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3)
```

### Description

Calculates Cohen's g and odds ratio for paired contingency tables, such as those that might be analyzed with McNemar or McNemar-Bowker tests.

### Usage

```r
cohenG(x, ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3)
```

### Arguments

- **x**: A two-way contingency table. It must be square. It can have two or more levels for each dimension.
- **ci**: If TRUE, returns confidence intervals by bootstrap. May be slow.
- **conf**: The level for the confidence interval.
- **type**: The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.
- **R**: The number of replications to use for bootstrap.
- **histogram**: If TRUE, produces a histogram of bootstrapped values.
- **digits**: The number of significant digits in the output.
Details

For a 2 x 2 table, where a and d are the concordant cells and b and c are discordant cells: Odds ratio is \( b/c \); \( P \) is \( b/(b+c) \); and Cohen’s \( g \) is \( P - 0.5 \).

In the 2 x 2 case, the statistics are directional. That is, when cell [1, 2] in the table is greater than cell [2, 1], OR is greater than 1, \( P \) is greater than 0.5, and \( g \) is positive.

In the opposite case, OR is less than 1, \( P \) is less than 0.5, and \( g \) is negative.

In the 2 x 2 case, when the effect is small, the confidence interval for OR can pass through 1, for \( g \) can pass through 0, and for \( P \) can pass through 0.5.

For tables larger than 2 x 2, the statistics are not directional. That is, OR is always \( \geq 1 \), \( P \) is always \( \geq 0.5 \), and \( g \) is always positive. Because of this, the confidence interval will never cross the values for no effect (OR = 1, \( P = 0.5 \), or \( g = 0 \)). Because of this, the confidence interval range should not be used for statistical inference for tables larger than 2 x 2.

When the reported statistics are close to their extremes, or with small counts, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A list containing: a data frame of results of the global statistics; and a data frame of results of the pairwise statistics.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_05.html

See Also

nominalSymmetryTest

Examples

```r
### 2 x 2 repeated matrix example
data(AndersonRainBarrel)
cohenG(AndersonRainBarrel)

### 3 x 3 repeated matrix
data(AndersonRainGarden)
cohenG(AndersonRainGarden)
```
cohenH

Cohen's h to compare proportions for 2 x 2 contingency tables

Description

Calculates Cohen's h for 2 x 2 contingency tables, such as those that might be analyzed with a chi-square test of association.

Usage

```r
cohenH(x, observation = "row", verbose = TRUE, digits = 3)
```

Arguments

- `x`: A 2 x 2 contingency table.
- `observation`: If "row", the row constitutes an observation. That is, the sum of each row is 100 percent. If "column", the column constitutes an observation. That is, the sum of each column is 100 percent.
- `verbose`: If TRUE, prints the proportions for each observation.
- `digits`: The number of significant digits in the output.

Details

Cohen's h is an effect size to compare two proportions. For a 2 x 2 table: Cohen's h equals \( \Phi_2 - \Phi_1 \), where, If observations are in rows, \( P_1 = \frac{a}{a+b} \) and \( P_2 = \frac{c}{c+d} \). If observations are in columns, \( P_1 = \frac{a}{a+c} \) and \( P_2 = \frac{b}{b+d} \). \( \Phi = 2 \times \arcsin(\sqrt{P}) \)

Value

A single statistic.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_05.html

See Also

cohenG

Examples

```r
data(Pennsylvania18)
Pennsylvania18
cohenH(Pennsylvania18, observation = "row")
```
cohenW

Cohen’s w (omega)

Description

Calculates Cohen’s w for a table of nominal variables.

Usage

cohenW(x, y = NULL, p = NULL, digits = 4, ...)

Arguments

x  Either a two-way table or a two-way matrix. Can also be a vector of observations for one dimension of a two-way table.
y  If x is a vector, y is the vector of observations for the second dimension of a two-way table.
p  If x is a vector of observed counts, p can be given as a vector of expected probabilities, as in a chi-square goodness of fit test.
digits  The number of significant digits in the output.
...  Additional arguments passed to chisq.test.

Details

Cohen’s w is used as a measure of association between two nominal variables, or as an effect size for a chi-square test of association. For a 2 x 2 table, the absolute value of the phi statistic is the same as Cohen’s w. The value of Cohen’s w is not bound by 1 on the upper end. Here, the value is always positive.

Value

A single statistic, Cohen’s w.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_10.html

See Also

cramerV cramervFfit
### Examples

#### Example with table
```r
data(Anderson)
fisher.test(Anderson)
cohenW(Anderson)
```

#### Example for goodness-of-fit
#### Bird foraging example, Handbook of Biological Statistics
```r
observed = c(70, 79, 3, 4)
expected = c(0.54, 0.40, 0.05, 0.01)
chisq.test(observed, p = expected)
cohenW(observed, p = expected)
```

#### Example with two vectors
```r
Species = c(rep("Species1", 16), rep("Species2", 16))
Color = c(rep(c("blue", "blue", "blue", "green"),4),
        rep(c("green", "green", "green", "blue"),4))
fisher.test(Species, Color)
cohenW(Species, Color)
```

---

**compareGLM**

**Compare fit statistics for glm models**

**Description**

Produces a table of fit statistics for multiple glm models.

**Usage**

```r
compareGLM(fits, ...)
```

**Arguments**

- **fits**: A series of model object names, separated by commas.
- **...**: Other arguments passed to `list`.

**Details**

Produces a table of fit statistics for multiple glm models: AIC, AICc, BIC, p-value, pseudo R-squared (McFadden, Cox and Snell, Nagelkerke).

Smaller values for AIC, AICc, and BIC indicate a better balance of goodness-of-fit of the model and the complexity of the model. The goal is to find a model that adequately explains the data without having too many terms.

BIC tends to choose models with fewer parameters relative to AIC. For comparisons with AIC, etc., to be valid, both models must have the same data, without transformations, use the same dependent variable, and be fit with the same method. They do not need to be nested.
Value

A list of two objects: The series of model calls, and a data frame of statistics for each model.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/rcompanion/e_07.html

See Also

compareLM, pairwiseModelAnova, accuracy

Examples

### Compare among logistic regression models

data(AndersonBias)
model.0 = glm(Result ~ 1, weight = Count, data = AndersonBias, 
  family = binomial(link="logit"))
model.1 = glm(Result ~ County, weight = Count, data = AndersonBias, 
  family = binomial(link="logit"))
model.2 = glm(Result ~ County + Sex, weight = Count, data = AndersonBias, 
  family = binomial(link="logit"))
model.3 = glm(Result ~ County + Sex + County:Sex, weight = Count, 
  data = AndersonBias, family = binomial(link="logit"))
compareGLM(model.0, model.1, model.2, model.3)

---

**compareLM**

*Compare fit statistics for lm models*

Description

Produces a table of fit statistics for multiple lm models.

Usage

```r
compareLM(fits, ...)
```

Arguments

- `fits`: A series of model object names, separated by commas.
- `...`: Other arguments passed to `list`. 
**Details**

Produces a table of fit statistics for multiple lm models: AIC, AICc, BIC, p-value, R-squared, and adjusted R-squared.

Smaller values for AIC, AICc, and BIC indicate a better balance of goodness-of-fit of the model and the complexity of the model. The goal is to find a model that adequately explains the data without having too many terms.

BIC tends to choose models with fewer parameters relative to AIC.

In the table, Shapiro.W and Shapiro.p are the W statistic and p-value for the Shapiro-Wilks test on the residuals of the model.

For comparisons with AIC, etc., to be valid, both models must have the same data, without transformations, use the same dependent variable, and be fit with the same method. They do not need to be nested.

**Value**

A list of two objects: The series of model calls, and a data frame of statistics for each model.

**Author(s)**

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

**References**


**See Also**

compareGLM, pairwiseModelAnova, accuracy

**Examples**

```r
### Compare among polynomial models
data(BrendonSmall)
BrendonSmall$Calories = as.numeric(BrendonSmall$Calories)

BrendonSmall$Calories2 = BrendonSmall$Calories * BrendonSmall$Calories
BrendonSmall$Calories3 = BrendonSmall$Calories * BrendonSmall$Calories * BrendonSmall$Calories
BrendonSmall$Calories4 = BrendonSmall$Calories * BrendonSmall$Calories * BrendonSmall$Calories * BrendonSmall$Calories

model.1 = lm(Sodium ~ Calories, data = BrendonSmall)
model.2 = lm(Sodium ~ Calories + Calories2, data = BrendonSmall)
model.3 = lm(Sodium ~ Calories + Calories2 + Calories3, data = BrendonSmall)
model.4 = lm(Sodium ~ Calories + Calories2 + Calories3 + Calories4, data = BrendonSmall)

compareLM(model.1, model.2, model.3, model.4)
```
Cramer’s V (phi)

Description

Calculates Cramer’s V for a table of nominal variables; confidence intervals by bootstrap.

Usage

cramerV(x, y = NULL, ci = FALSE, conf = 0.95, type = "perc",
R = 1000, histogram = FALSE, digits = 4, bias.correct = FALSE,
...)

Arguments

x
Either a two-way table or a two-way matrix. Can also be a vector of observations for one dimension of a two-way table.

y
If x is a vector, y is the vector of observations for the second dimension of a two-way table.

cl
If TRUE, returns confidence intervals by bootstrap. May be slow.

conf
The level for the confidence interval.

type
The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.

R
The number of replications to use for bootstrap.

histogram
If TRUE, produces a histogram of bootstrapped values.

digits
The number of significant digits in the output.

bias.correct
If TRUE, a bias correction is applied.

... Additional arguments passed to chisq.test.

Details

Cramer’s V is used as a measure of association between two nominal variables, or as an effect size for a chi-square test of association. For a 2 x 2 table, the absolute value of the phi statistic is the same as Cramer’s V.

Because V is always positive, the confidence interval will never cross zero. The confidence interval range should not be used for statistical inference.

When V is close to 0 or very large, or with small counts, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, Cramer’s V. Or a small data frame consisting of Cramer’s V, and the lower and upper confidence limits. In the data frame, r is just a duplicate of the Cramer’s V value.
Author(s)
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References
http://rcompanion.org/handbook/H_10.html

See Also
cohenW

Examples
### Example with table
data(Anderson)
fisher.test(Anderson)
cramerV(Anderson)

### Example with two vectors
Species = c(rep("Species1", 16), rep("Species2", 16))
Color = c(rep(c("blue", "blue", "blue", "green"),4),
         rep(c("green", "green", "green", "blue"),4))
fisher.test(Species, Color)
cramerV(Species, Color)

---

**cramerVFit**  
*Cramer’s V for chi-square goodness-of-fit tests*

**Description**
Calculates Cramer’s V for a vector of counts and expected counts; confidence intervals by bootstrap.

**Usage**
cramerVFit(x, p = rep(1/length(x), length(x)), ci = FALSE,
conf = 0.95, type = "perc", R = 1000, histogram = FALSE,
digits = 4, ...)

**Arguments**
- **x**  
  A vector of observed counts.
- **p**  
  A vector of expected or default probabilities.
- **ci**  
  If TRUE, returns confidence intervals by bootstrap. May be slow.
- **conf**  
  The level for the confidence interval.
- **type**  
  The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.
The number of replications to use for bootstrap.

If TRUE, produces a histogram of bootstrapped values.

The number of significant digits in the output.

Additional arguments passed to `chisq.test`.

Details

This modification of Cramer's V could be used to indicate an effect size in cases where a chi-square goodness-of-fit test might be used. It indicates the degree of deviation of observed counts from the expected probabilities.

In the case of equally-distributed expected frequencies, Cramer's V will be equal to 1 when all counts are in one category, and it will be equal to 0 when the counts are equally distributed across categories. This does not hold if the expected frequencies are not equally-distributed.

Because V is always positive, the confidence interval will never cross zero. The confidence interval range should not be used for statistical inference.

When V is close to 0 or 1, or with small counts, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, Cramer's V. Or a small data frame consisting of Cramer's V, and the lower and upper confidence limits. In the data frame, \( r \) is just a duplicate of the Cramer's V value.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_03.html

See Also

cramerV

Examples

```r
### Equal probabilities example
### From http://rcompanion.org/handbook/H_03.html
nail.color = c("Red", "None", "White", "Green", "Purple", "Blue")
observed = c( 19, 3, 1, 1, 2, 2 )
expected = c( 1/6, 1/6, 1/6, 1/6, 1/6, 1/6 )
chisq.test(x = observed, p = expected)
cramerVFit(x = observed, p = expected)

### Unequal probabilities example
### From http://rcompanion.org/handbook/H_03.html
race = c("White", "Black", "American Indian", "Asian", "Pacific Islander", "Two or more races")
```
epsilonSquared

\[
\begin{align*}
\text{observed} &= c(20, 9, 9, 1, 1, 1) \\
\text{expected} &= c(0.775, 0.132, 0.012, 0.054, 0.002, 0.025) \\
\text{chisq.test(x = observed, p = expected)} \\
\text{cramerVFit(x = observed, p = expected)}
\end{align*}
\]

### Examples of perfect and zero fits
\[
\begin{align*}
\text{cramerVFit(c(100, 0, 0, 0, 0))} \\
\text{cramerVFit(c(10, 10, 10, 10, 10))}
\end{align*}
\]

---

**epsilonSquared**  
*Epsilon-squared*

**Description**

Calculates epsilon-squared for a table with one ordinal variable and one nominal variable; confidence intervals by bootstrap.

**Usage**

\[
\text{epsilonSquared(x, g = NULL, group = "row", ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3, ...)}
\]

**Arguments**

- **x**: Either a two-way table or a two-way matrix. Can also be a vector of observations of an ordinal variable.
- **g**: If `x` is a vector, `g` is the vector of observations for the grouping, nominal variable.
- **group**: If `x` is a table or matrix, `group` indicates whether the "row" or the "column" variable is the nominal, grouping variable.
- **ci**: If TRUE, returns confidence intervals by bootstrap. May be slow.
- **conf**: The level for the confidence interval.
- **type**: The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to `boot.ci`.
- **R**: The number of replications to use for bootstrap.
- **histogram**: If TRUE, produces a histogram of bootstrapped values.
- **digits**: The number of significant digits in the output.
- **...**: Additional arguments passed to the `kruskal.test` function.
Details

Epsilon-squared is used as a measure of association for the Kruskal-Wallis test or for a two-way table with one ordinal and one nominal variable.

Currently, the function makes no provisions for NA values in the data. It is recommended that NAs be removed beforehand.

Because epsilon-squared is always positive, the confidence interval will never cross zero. The confidence interval range should not be used for statistical inference.

When epsilon-squared is close to 0 or very large, or with small counts in some cells, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, epsilon-squared. Or a small data frame consisting of epsilon-squared, and the lower and upper confidence limits.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_11.html

See Also

freemanTheta

Examples

data(Breakfast)
library(coin)
chisq_test(Breakfast, scores = list("Breakfast" = c(-2, -1, 0, 1, 2)))
epsilonSquared(Breakfast)

data(PoohPiglet)
kruskal.test(Likert ~ Speaker, data = PoohPiglet)
epsilonSquared(x = PoohPiglet$Likert, g = PoohPiglet$Speaker)

### Same data, as matrix of counts

data(PoohPiglet)
XT = xtabs(~ Speaker + Likert, data = PoohPiglet)
epsilonSquared(XT)
Description

Calculates Freeman’s theta for a table with one ordinal variable and one nominal variable; confidence intervals by bootstrap.

Usage

freemanTheta(x, g = NULL, group = "row", verbose = FALSE, progress = FALSE, ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3)

Arguments

x Either a two-way table or a two-way matrix. Can also be a vector of observations of an ordinal variable.
g If x is a vector, g is the vector of observations for the grouping, nominal variable.
group If x is a table or matrix, group indicates whether the "row" or the "column" variable is the nominal, grouping variable.
verbose If TRUE, prints statistics for each comparison.
progress If TRUE, prints a message as each comparison is conducted.
ci If TRUE, returns confidence intervals by bootstrap. May be slow.
conf The level for the confidence interval.
type The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.
R The number of replications to use for bootstrap.
histogram If TRUE, produces a histogram of bootstrapped values.
digits The number of significant digits in the output.

Details

Freeman’s coefficient of differentiation (theta) is used as a measure of association for a two-way table with one ordinal and one nominal variable. See Freeman (1965).

Currently, the function makes no provisions for NA values in the data. It is recommended that NAs be removed beforehand.

Because theta is always positive, the confidence interval will never cross zero. The confidence interval range should not be used for statistical inference.

When theta is close to 0 or very large, or with small counts in some cells, the confidence intervals determined by this method may not be reliable, or the procedure may fail.
Value

A single statistic, Freeman’s theta. Or a small data frame consisting of Freeman’s theta, and the lower and upper confidence limits.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References


http://rcompanion.org/handbook/H_11.html

See Also

epsilonSquared

Examples

data(Breakfast)
library(coin)
chisq_test(Breakfast, scores = list("Breakfast" = c(-2, -1, 0, 1, 2)))
freemanTheta(Breakfast)

### Example from Freeman (1965), Table 10.6
Counts = c(1,2,5,2,0,10,5,0,0,0,2,2,1,0,0,0,2,3)
Matrix = matrix(Counts, byrow=TRUE, ncol=5,
dimnames = list(Marital.status=c("Single","Married","Widowed","Divorced"),
Social.adjustment = c("5","4","3","2","1")))
Matrix
freemanTheta(Matrix)

### Example after Kruskal Wallis test
data(PoohPiglet)
kruskal.test(Likert ~ Speaker, data = PoohPiglet)
freemanTheta(x = PoohPiglet$Likert, g = PoohPiglet$Speaker)

### Same data, as table of counts
data(PoohPiglet)
XT = xtabs(~ Speaker + Likert , data = PoohPiglet)
freemanTheta(XT)
### fullPTable

**Convert a lower triangle matrix to a full matrix**

**Description**

Converts a lower triangle matrix to a full matrix.

**Usage**

```r
fullPTable(PT)
```

**Arguments**

- `PT` A lower triangle matrix.

**Details**

This function is useful to convert a lower triangle matrix of p-values from a pairwise test to a full matrix. A full matrix can be passed to `multcompLetters` in the `multcompView` package to produce a compact letter display.

**Value**

A full matrix.

**Author(s)**

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

**References**

http://rcompanion.org/handbook/F_08.html

**Examples**

```r
### Example with pairwise.wilcox.test
data(BrendonSmall)
BrendonSmall$Instructor = factor(BrendonSmall$Instructor,
                                 levels = c('Brendon Small', 'Jason Penopolis',
                                            'Paula Small', 'Melissa Robbins',
                                            'Coach McGuirk'))
P = pairwise.wilcox.test(x = BrendonSmall$Score, g = BrendonSmall$Instructor)
PT = P$p.value
PT
PT1 = fullPTable(PT)
PT1
library(multcompView)
multcompLetters(PT1)
```
groupwiseCMH

Post-hoc tests for Cochran-Mantel-Haenszel test

Description

Conducts groupwise tests of association on a three-way contingency table.

Usage

```r
groupwiseCMH(x, group = 3, fisher = TRUE, gtest = FALSE,
chisq = FALSE, method = "fdr", correct = "none", digits = 3, ...)
```

Arguments

- `x`: A three-way contingency table.
- `group`: The dimension of the table to use as the grouping variable. Will be 1, 2, or 3.
- `fisher`: If TRUE, conducts Fisher exact test.
- `gtest`: If TRUE, conducts G test of association.
- `chisq`: If TRUE, conducts Chi-square test of association.
- `method`: The method to use to adjust p-values. See `?p.adjust`.
- `correct`: The correction to apply to the G test. See `GTest`.
- `digits`: The number of digits for numbers in the output.
- `...`: Other arguments passed to `chisq.test` or `GTest`.

Details

If more than one of `fisher`, `gtest`, or `chisq` is set to TRUE, only one type of test of association will be conducted.

Value

A data frame of groups, test used, p-values, and adjusted p-values.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_06.html

See Also

`nominalSymmetryTest`, `pairwiseMcnemar`, `pairwiseNominalIndependence`, `pairwiseNominalMatrix`
Examples

```r
### Post-hoc for Cochran-Mantel-Haenszel test
data(AndersonBias)
Table = xtabs(Count ~ Sex + Result + County, 
data=AndersonBias)
ftable(Table)
mantelhaen.test(Table)
groupwiseCMH(Table, 
group = 3,
fisher = TRUE, 
gtest = FALSE, 
chisq = FALSE, 
method = "fdr", 
correct = "none", 
digits = 3)
```

---

**groupwiseGeometric**  
*Groupwise geometric means and confidence intervals*

**Description**

Calculates geometric means and confidence intervals for groups.

**Usage**

```r
groupwiseGeometric(formula = NULL, data = NULL, var = NULL, 
                  group = NULL, conf = 0.95, na.rm = TRUE, digits = 3, ...)
```

**Arguments**

- `formula`: A formula indicating the measurement variable and the grouping variables. e.g. `y ~ x1 + x2`.
- `data`: The data frame to use.
- `var`: The measurement variable to use. The name is in double quotes.
- `group`: The grouping variable to use. The name is in double quotes. Multiple names are listed as a vector. (See example.)
- `conf`: The confidence interval to use.
- `na.rm`: If TRUE, removes NA values in the measurement variable.
- `digits`: The number of significant figures to use in output.
- `...`: Other arguments. Not currently useful.
The input should include either formula and data; or data, var, and group. (See examples).

The function computes means, standard deviations, standard errors, and confidence intervals on log-transformed values. Confidence intervals are calculated in the traditional manner with the t-distribution. These statistics assume that the data are log-normally distributed. For data not meeting this assumption, medians and confidence intervals by bootstrap may be more appropriate.

A data frame of geometric means, standard deviations, standard errors, and confidence intervals.

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The variables on the right side are used for the grouping variables.

Results for ungrouped (one-sample) data can be obtained by either setting the right side of the formula to 1, e.g. y ~ 1, or by setting group=NULL.

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

http://rcompanion.org/handbook/C_03.html

Examples

```r
## Example with formula notation
data(Catbus)
groupwiseGeometric(Steps ~ Sex + Teacher,
data = Catbus)

## Example with variable notation
data(Catbus)
groupwiseGeometric(data = Catbus,
var = "Steps",
group = c("Sex", "Teacher"))
```
groupwiseHuber  

*Groupwise Huber M-estimators and confidence intervals*

**Description**

Calculates Huber M-estimator and confidence intervals for groups.

**Usage**

```r
groupwiseHuber(formula = NULL, data = NULL, var = NULL, 
    group = NULL, conf.level = 0.95, ci.type = "wald", ...)
```

**Arguments**

- `formula`
  - A formula indicating the measurement variable and the grouping variables. e.g. `y ~ x1 + x2`.
- `data`
  - The data frame to use.
- `var`
  - The measurement variable to use. The name is in double quotes.
- `group`
  - The grouping variable to use. The name is in double quotes. Multiple names are listed as a vector. (See example.)
- `conf.level`
  - The confidence interval to use.
- `ci.type`
  - The type of confidence interval to use. Can be "wald" or "boot". See `HuberM` for details.
- `...`
  - Other arguments passed to the `HuberM` function.

**Details**

A wrapper for the `DescTools::HuberM` function to allow easy output for multiple groups.

The input should include either `formula` and `data`; or `data`, `var`, and `group`. (See examples).

**Value**

A data frame of requested statistics by group.

**Note**

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The variables on the right side are used for the grouping variables.

Results for ungrouped (one-sample) data can be obtained by either setting the right side of the formula to 1, e.g. `y ~ 1`, or by setting `group=NULL`.

**Author(s)**

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>
References

http://rcompanion.org/rcompanion/d_08a.html

See Also

groupwiseMean, groupwiseMedian, groupwiseGeometric

Examples

```r
### Example with formula notation
data(Catbus)
groupwiseHuber(Steps ~ Teacher + Sex, 
data = Catbus, 
   ci.type = "wald")

### Example with variable notation
data(Catbus)
groupwiseHuber(data = Catbus, 
   var = "Steps", 
   group = c("Teacher", "Sex"), 
   ci.type = "wald")
```

---

**groupwiseMean**  
*Groupwise means and confidence intervals*

**Description**

Calculates means and confidence intervals for groups.

**Usage**

```r
groupwiseMean(formula = NULL, data = NULL, var = NULL, 
  group = NULL, conf = 0.95, R = 5000, boot = FALSE, 
  traditional = TRUE, normal = FALSE, basic = FALSE, 
  percentile = FALSE, bca = FALSE, digits = 3, ...)
```

**Arguments**

- **formula**  
  A formula indicating the measurement variable and the grouping variables. e.g.  
  `y ~ x1 + x2`.
- **data**  
  The data frame to use.
- **var**  
  The measurement variable to use. The name is in double quotes.
- **group**  
  The grouping variable to use. The name is in double quotes. Multiple names are listed as a vector. (See example.)
- **conf**  
  The confidence interval to use.
- **R**  
  The number of bootstrap replicates to use for bootstrapped statistics.
**groupwiseMean**

- **boot** If TRUE, includes the mean of the bootstrapped means. This can be used as an estimate of the mean for the group.
- **traditional** If TRUE, includes the traditional confidence intervals for the group means, using the t-distribution.
- **normal** If TRUE, includes the normal confidence intervals for the group means by bootstrap. See `boot.ci`.
- **basic** If TRUE, includes the basic confidence intervals for the group means by bootstrap. See `boot.ci`.
- **percentile** If TRUE, includes the percentile confidence intervals for the group means by bootstrap. See `boot.ci`.
- **bca** If TRUE, includes the BCa confidence intervals for the group means by bootstrap. See `boot.ci`.
- **digits** The number of significant figures to use in output.
- **...** Other arguments passed to the `boot` function.

**Details**

The input should include either `formula` and `data`, or `data`, `var`, and `group`. (See examples).

With some options, the function may not handle missing values well. This seems to happen particularly with `bca = TRUE`.

**Value**

A data frame of requested statistics by group.

**Note**

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The variables on the right side are used for the grouping variables.

Results for ungrouped (one-sample) data can be obtained by either setting the right side of the formula to 1, e.g. `y ~ 1`, or by setting `group=NULL`.

**Author(s)**

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

**References**

http://rcompanion.org/handbook/C_03.html

**See Also**

`groupwiseMedian`, `groupwiseHuber`, `groupwiseGeometric`
Examples

### Example with formula notation
```r
data(Catbus)
groupwiseMean(Steps ~ Teacher + Sex, 
data = Catbus, 
  traditional = FALSE, 
  percentile = TRUE)
```

### Example with variable notation
```r
data(Catbus)
groupwiseMean(data = Catbus, 
  var = "Steps", 
  group = c("Teacher", "Sex"), 
  traditional = FALSE, 
  percentile = TRUE)
```

groupwiseMedian

Groupwise medians and confidence intervals

Description

Calculates medians and confidence intervals for groups.

Usage

```r
groupwiseMedian(formula = NULL, data = NULL, var = NULL, 
group = NULL, conf = 0.95, R = 5000, boot = FALSE, 
pseudo = FALSE, basic = FALSE, normal = FALSE, 
percentile = FALSE, bca = TRUE, wilcox = FALSE, exact = FALSE, 
digits = 3, ...)
```

Arguments

- `formula`: A formula indicating the measurement variable and the grouping variables. e.g. `y ~ x1 + x2`.
- `data`: The data frame to use.
- `var`: The measurement variable to use. The name is in double quotes.
- `group`: The grouping variable to use. The name is in double quotes. Multiple names are listed as a vector. (See example.)
- `conf`: The confidence interval to use.
- `R`: The number of bootstrap replicates to use for bootstrapped statistics.
- `boot`: If `TRUE`, includes the mean of the bootstrapped medians. This can be used as an estimate of the median for the group.
- `pseudo`: If `TRUE`, includes the pseudo median from `wilcox.test`. 


groupwiseMedian

basic
If TRUE, includes the basic confidence intervals for the group means by bootstrap. See \texttt{boot::boot.ci}.

normal
If TRUE, includes the normal confidence intervals for the group means by bootstrap. See \texttt{boot::boot.ci}.

percentile
If TRUE, includes the percentile confidence intervals for the group means by bootstrap. See \texttt{boot::boot.ci}.

bca
If TRUE, includes the BCa confidence intervals for the group means by bootstrap. See \texttt{boot::boot.ci}.

wilcox
If TRUE, includes the wilcox confidence intervals from \texttt{stats::wilcox.test}.

exact
If TRUE, includes the "exact" confidence intervals from \texttt{DescTools::MedianCI}.

digits
The number of significant figures to use in output.

... Other arguments passed to the \texttt{boot} function.

Details

The input should include either \texttt{formula} and \texttt{data}; or \texttt{data}, \texttt{var}, and \texttt{group}. (See examples).

With some options, the function may not handle missing values well. This seems to happen particularly with \texttt{bca = TRUE}.

Value

A data frame of requested statistics by group.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The variables on the right side are used for the grouping variables.

Results for ungrouped (one-sample) data can be obtained by either setting the right side of the formula to 1, e.g. \texttt{y ~ 1}, or by setting \texttt{group=NULL}.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/E_04.html

See Also

groupwiseMean, groupwiseHuber groupwiseGeometric
Examples

```r
### Example with formula notation
data(Catbus)
groupwiseMedian(Steps ~ Teacher + Sex,
data = Catbus,
bc = FALSE,
percentile = TRUE,
R = 1000)
```

```r
### Example with variable notation
data(Catbus)
groupwiseMedian(data = Catbus,
var = "Steps",
group = c("Teacher", "Sex"),
bc = FALSE,
percentile = TRUE,
R = 1000)
```

---

**groupwisePercentile**

*Groupwise percentiles and confidence intervals*

**Description**

Calculates percentiles and confidence intervals for groups.

**Usage**

```r
groupwisePercentile(formula = NULL, data = NULL, var = NULL,
group = NULL, conf = 0.95, tau = 0.5, type = 7, R = 5000,
boot = FALSE, basic = FALSE, normal = FALSE, percentile = FALSE,
bc = TRUE, digits = 3, ...)
```

**Arguments**

- `formula` A formula indicating the measurement variable and the grouping variables. e.g. `y ~ x1 + x2`.
- `data` The data frame to use.
- `var` If no formula is given, the measurement variable to use. The name is in double quotes.
- `group` The grouping variable to use. The name is in double quotes. Multiple names are listed as a vector. (See example.)
- `conf` The confidence interval to use.
- `tau` The percentile to use, expressed as a quantile, e.g. 0.5 for median, 0.25 for 25th percentile.
The input should include either `formula` and `data`; or `data`, `var`, and `group`. (See examples).

With some options, the function may not handle missing values well. This seems to happen particularly with `bca=TRUE`.

### Value

A data frame of requested statistics by group

### Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The variables on the right side are used for the grouping variables.

Results for ungrouped (one-sample) data can be obtained by either setting the right side of the formula to 1, e.g. `y ~ 1`, or by setting `group=NULL`.

### Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

### References

http://rcompanion.org/handbook/F_15.html

### See Also

`groupwiseMean`, `groupwiseHuber`, `groupwiseGeometric`, `groupwiseMedian`
Examples

### Example with formula notation
```r
data(Catbus)
groupwisePercentile(Steps ~ Teacher + Sex, 
data = Catbus, 
tau = 0.25, 
bcv = FALSE, 
percentile = TRUE, 
R = 1000)
```

### Example with variable notation
```r
data(Catbus)
groupwisePercentile(data = Catbus, 
var = "Steps", 
group = c("Teacher", "Sex"), 
tau = 0.25, 
bcv = FALSE, 
percentile = TRUE, 
R = 1000)
```

groupwiseSum  

Groupwise sums

Description

Calculates sums for groups.

Usage

```r
groupwiseSum(formula = NULL, data = NULL, var = NULL, group = NULL, 
digits = NULL, ...)
```

Arguments

- **formula**: A formula indicating the measurement variable and the grouping variables. e.g. \(y \sim x1 + x2\).
- **data**: The data frame to use.
- **var**: The measurement variable to use. The name is in double quotes.
- **group**: The grouping variable to use. The name is in double quotes. Multiple names are listed as a vector. (See example.)
- **digits**: The number of significant figures to use in output. The default is NULL, which results in no rounding of values.
- **...**: Other arguments passed to the `sum` function

Details

The input should include either `formula` and `data`; or `data`, `var`, and `group`. (See examples).
Description
A data frame in long form with yes/no responses for four lawn care practices for each of 14 respondents. Hypothetical data.

Usage
HayleySmith

Format
An object of class data.frame with 56 rows and 3 columns.
**kendallW**

**Kendall’s W with bootstrapped confidence interval**

**Description**

Calculates Kendall’s W coefficient of concordance, which can be used as an effect size statistic for unreplicated complete block design such as where Friedman’s test might be used. This function is a wrapper for the `KendallW` function in the `DescTools` package, with the addition of bootstrapped confidence intervals.

**Usage**

```r
kendallW(x, correct = TRUE, na.rm = FALSE, ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3, ...)
```

**Arguments**

- `x` A k x m matrix or table, with k treatments in rows and m raters or blocks in columns.
- `correct` Passed to `KendallW`.
- `na.rm` Passed to `KendallW`.
- `ci` If TRUE, returns confidence intervals by bootstrap. May be slow.
- `conf` The level for the confidence interval.
- `type` The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to `boot.ci`.
- `R` The number of replications to use for bootstrap.
- `histogram` If TRUE, produces a histogram of bootstrapped values.
- `digits` The number of significant digits in the output.
- `...` Additional arguments passed to the `KendallW` function.

**Details**

See the `KendallW` function in the `DescTools` package for details.

When W is close to 0 or very large, or with small sample size, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Because W is always positive, the confidence interval will never cross zero. The confidence interval range should not be used for statistical inference.

When producing confidence intervals by bootstrap, this function treats each rater or block as an observation. It is not clear to the author if this approach produces accurate confidence intervals, but it appears to be reasonable.
Monarchs

Value

A single statistic, W. Or a small data frame consisting of W, and the lower and upper confidence limits.

Acknowledgments

Thanks to Indrajeet Patil, author of ggstatsplot, and groupedstats for help in the inspiring and coding of this function.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_04.html

Examples

data(BobBelcher)
Table = xtabs(Likert ~ Instructor + Rater, data = BobBelcher)
kendallW(Table)

---

Monarchs

*Hypothetical data for monarch butterflies in gardens*

Description

A data frame of the number of monarch butterflies in three gardens. Hypothetical data.

Usage

Monarchs

Format

An object of class data.frame with 24 rows and 2 columns.

Source

http://rcompanion.org/handbook/J_01.html
multiVDA

Pairwise Vargha and Delaney’s A and Cliff’s delta

Description
Calculates Vargha and Delaney’s A (VDA), Cliff’s delta (CD), and r for several groups in a pairwise manner.

Usage
multiVDA(formula = NULL, data = NULL, x = NULL, g = NULL, 
statistic = "VDA", digits = 3, ...)

Arguments
- formula: A formula indicating the response variable and the independent variable. E.g. y ~ group.
- data: The data frame to use.
- x: If no formula is given, the response variable.
- g: If no formula is given, the grouping variable.
- statistic: One of "VDA", "CD", or "r". This determines which statistic will be evaluated to determine the comparison with the most divergent groups.
- digits: The number of significant digits in the output.
- ...: Additional arguments passed to the wilcox.test function.

Details
VDA and CD are effect size statistic appropriate in cases where a Wilcoxon-Mann-Whitney test might be used. Here, the pairwise approach would be used in cases where a Kruskal-Wallis test might be used. VDA ranges from 0 to 1, with 0.5 indicating stochastic equality, and 1 indicating that the first group dominates the second. CD ranges from -1 to 1, with 0 indicating stochastic equality, and 1 indicating that the first group dominates the second. r ranges from approximately, -0.86 to 0.86, depending on sample size, with 0 indicating no effect, and a positive result indicating that values in the first group are greater than in the second.

In the function output, VDA.m is the greater of VDA or 1-VDA. CD.m is the absolute value of CD. r.m is the absolute value of r.

The function calculates VDA and Cliff’s delta from the "W" U statistic from the wilcox.test function. Specifically, VDA = U/(n1*n2); CD = (VDA-0.5)+2. For r, the Z value is extracted from the wilcox_test function in the coin package. r is calculated as Z divided by square root of the total observations.

The input should include either formula and data; or var, and group.

Currently, the function makes no provisions for NA values in the data. It is recommended that NAs be removed beforehand.
When the data in the first group are greater than in the second group, VDA is > 0.5, CD is positive, and r is positive. When the data in the second group are greater than in the first group, VDA is < 0.5, CD is negative, and r is negative. Be cautious with this interpretation, as R will alphabetize groups in the formula interface if the grouping variable is not already a factor.

Value

A list containing a data frame of pairwise statistics, and the comparison with the most extreme value of the chosen statistic.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_08.html

See Also

cliffDelta, cliffDelta

Examples

data(PoohPiglet)
multiVDA(Likert ~ Speaker, data=PoohPiglet)

nagelkerke  

nagelkerke(fit, null = NULL, restrictNobs = FALSE)

description

Produces McFadden, Cox and Snell, and Nagelkerke pseudo R-squared measures, along with p-values, for models.

Usage

nagelkerke(fit, null = NULL, restrictNobs = FALSE)
nagelkerke

Arguments

- **fit**: The fitted model object for which to determine pseudo r-squared.
- **null**: The null model object against which to compare the fitted model object. The null model must be nested in the fitted model to be valid. Specifying the null is optional for some model object types and is required for others.

- **restrictNobs**: If TRUE, limits the observations for the null model to those used in the fitted model. Works with only some model object types.

Details

Pseudo R-squared values are not directly comparable to the R-squared for OLS models. Nor can they be interpreted as the proportion of the variability in the dependent variable that is explained by model. Instead pseudo R-squared measures are relative measures among similar models indicating how well the model explains the data.

Cox and Snell is also referred to as ML. Nagelkerke is also referred to as Cragg and Uhler.

Model objects accepted are lm, glm, gls, lme, lmer, lmerTest, nls, clm, clmm, vglm, glmer, negbin, zero-infl, betareg, and rq.

Model objects that require the null model to be defined are nls, lmer, glmer, and clmm. Other objects use the `update` function to define the null model.

Likelihoods are found using ML (REML = FALSE).

The fitted model and the null model should be properly nested. That is, the terms of one need to be a subset of the other, and they should have the same set of observations. One issue arises when there are NA values in one variable but not another, and observations with NA are removed in the model fitting. The result may be fitted and null models with different sets of observations.

Setting `restrictNobs` to TRUE ensures that only observations in the fit model are used in the null model. This appears to work for lm and some glm models, but causes the function to fail for other model object types.

Some pseudo R-squared measures may not be appropriate or useful for some model types.

Calculations are based on log likelihood values for models. Results may be different than those based on deviance.

Value

A list of six objects describing the models used, the pseudo r-squared values, the likelihood ratio test for the model, the number of obervaton for the models, messages, and any warnings.

Acknowledgements

My thanks to Jan-Herman Kuiper of Keele University for suggesting the `restrictNobs` fix.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/G_10.html
nagelkerkeHermite

See Also

accuracy

Examples

### Logistic regression example
```r
data(AndersonBias)
model = glm(Result ~ County + Sex + County:Sex,
            weight = Count,
            data = AndersonBias,
            family = binomial(link="logit"))
nagelkerke(model)
```

### Quadratic plateau example
### With nls, the null needs to be defined
```r
data(BrendonSmall)
quadplat = function(x, a, b, clx) {
    ifelse(x < clx, a + b * x + (-0.5*b/clx) * x * x,
           a + b * clx + (-0.5*b/clx) * clx * clx)
}
model = nls(Sodium ~ quadplat(Calories, a, b, clx),
            data = BrendonSmall,
            start = list(a = 519,
                          b = 0.359,
                          clx = 2304))
nullfunct = function(x, m){m}
null.model = nls(Sodium ~ nullfunct(Calories, m),
                data = BrendonSmall,
                start = list(m = 1346))
nagelkerke(model, null=null.model)
```

---

nagelkerkeHermite  [Defunct!] Pseudo r-squared measures for hermite models

Description

Defunct. Produces McFadden, Cox and Snell, and Nagelkerke pseudo R-squared measures, along with p-value for the model, for hermite regression objects.

Usage

nagelkerkeHermite(...)

Arguments

... Anything.
nominalSymmetryTest

*Exact and Monte Carlo symmetry tests for paired contingency tables*

**Description**

Conducts an omnibus symmetry test for a paired contingency table and then post-hoc pairwise tests. This is similar to McNemar and McNemar-Bowker tests in use.

**Usage**

nominalSymmetryTest(x, method = "fdr", digits = 3, ...)

**Arguments**

- `x` A two-way contingency table. It must be square. It can have two or more levels for each dimension.
- `method` The method to adjust multiple p-values. See `stats::p.adjust`.
- `digits` The number of significant digits in the output.
- `...` Additional arguments, passed to `EMT::multinomial.test`.

**Details**

If Monte Carlo is not used, the test of symmetry uses an exact test by conducting either a binomial or multinomial goodness-of-fit test.

These are equivalent to uncorrected McNemar and McNemar-Bowker tests, but will not fail when there are zeros in critical cells, as will the `mcnemar.test` function.

**Value**

A list containing: a data frame of results of the global test; a data frame of results of the pairwise results; and a data frame mentioning the p-value adjustment method.

**Author(s)**

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

**References**

http://rcompanion.org/handbook/H_05.html

**See Also**

`pairwiseMcnemar`, `groupwiseCMH`, `pairwiseNominalIndependence`, `pairwiseNominalMatrix`
Examples

### 2 x 2 repeated matrix example
```r
data(AndersonRainBarrel)
nominalSymmetryTest(AndersonRainBarrel)
```

### 3 x 3 repeated matrix example with Monte Carlo
```r
data(AndersonRainGarden)
nominalSymmetryTest(AndersonRainGarden,  
                         MonteCarlo = TRUE,  
                         ntrial = 10000)
```

### 4 x 4 repeated matrix example that fails with mcnemar.test
```r
data(Religion)
nominalSymmetryTest(Religion,  
                        MonteCarlo = TRUE,  
                        ntrial = 10000)
```

---

### Nurseries

Data for proportion of good practices followed by plant nurseries

**Description**

A data frame with two variables: size of plant nursery in hectares, and proportion of good practices followed by the nursery

**Usage**

Nurseries

**Format**

An object of class `data.frame` with 38 rows and 2 columns.

**Source**

pairwiseDifferences

[Defunct!] Pairwise differences for unreplicated CBD

Description

Defunct. Calculates the differences in the response variable for each pair of levels of a grouping variable in an unreplicated complete block design.

Usage

pairwiseDifferences(...)

Arguments

... Anything.

pairwiseMcNemar Pairwise McNemar and related tests for Cochran Q test post-hoc

Description

Conducts pairwise McNemar, exact, and permutation tests as a post-hoc to Cochran Q test.

Usage

pairwiseMcNemar(formula = NULL, data = NULL, x = NULL, g = NULL, block = NULL, test = "exact", method = "fdr", digits = 3, correct = FALSE)

Arguments

formula A formula indicating the measurement variable and the grouping variable. e.g. y ~ group | block.
data The data frame to use.
x The response variable.
g The grouping variable.
block The blocking variable.
test If "exact", conducts an exact test of symmetry analogous to a McNemar test. If "mcnemar", conducts a McNemar test of symmetry. If "permutation", conducts a permutation test analogous to a McNemar test.
method The method for adjusting multiple p-values. See p.adjust.
digits The number of significant digits in the output.
correct If TRUE, applies a continuity correction for the McNemar test.
pairwiseMcnemar

Details

The component tables for the pairwise tests must be of size 2 x 2.
The input should include either formula and data; or x, g, and block.

Value

A list containing: a data frame of results of the global test; a data frame of results of the pairwise results; and a data frame mentioning the p-value adjustment method.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable. The second variable on the right side is used for the blocking variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_07.html

See Also

nominalSymmetryTest, groupwiseCMH, pairwiseNominalIndependence, pairwiseNominalMatrix

Examples

### Cochran Q post-hoc example

data(HayleySmith)
library(DescTools)
CochranQTest(Response ~ Practice | Student,
data = HayleySmith)
HayleySmith$Practice = factor(HayleySmith$Practice,
levels = c("MowHeight", "SoilTest", "Clippings", "Irrigation"))
PT = pairwiseMcnemar(Response ~ Practice | Student,
data = HayleySmith,
test = "exact",
method = "fdr",
digits = 3)

PT
PT = PT$Pairwise
cldList(comparison = PT$Comparison,
p.value = PT$p.adjust,
threshold = 0.05)
pairwiseMedianMatrix

Pairwise Mood’s median tests with matrix output

Description
Conducts pairwise Mood’s median tests across groups.

Usage
pairwiseMedianMatrix(formula = NULL, data = NULL, x = NULL, g = NULL, digits = 4, method = "fdr", ...)

Arguments
formula A formula indicating the measurement variable and the grouping variable. e.g. y ~ group.
data The data frame to use.
x The response variable as a vector.
g The grouping variable as a vector.
digits The number of significant digits to round output.
method The p-value adjustment method to use for multiple tests. See stats::p.adjust.
... Additional arguments passed to coin::median_test.

Details
The input should include either formula and data; or x, and g.
Mood’s median test compares medians among two or more groups. See http://rcompanion.org/handbook/F_09.html for further discussion of this test.
The pairwiseMedianMatrix function can be used as a post-hoc method following an omnibus Mood’s median test. The matrix output can be converted to a compact letter display.

Value
A list consisting of: a matrix of p-values; the p-value adjustment method; a matrix of adjusted p-values.

Note
The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>
pairwiseMedianTest

References
http://rcompanion.org/handbook/F_09.html

See Also
pairwiseMedianTest

Examples

data(PoohPiglet)
PoohPiglet$Speaker = factor(PoohPiglet$Speaker,
levels = c("Pooh", "Tigger", "Piglet"))
PT = pairwiseMedianMatrix(Likert ~ Speaker,
data = PoohPiglet,
exact = NULL,
method = "fdr")$Adjusted
PT
library(multcompView)
multcompLetters(PT,
compare="<",
threshold=0.05,
Letters=letters)

pairwiseMedianTest  Pairwise Mood’s median tests

Description
Conducts pairwise Mood’s median tests across groups.

Usage
pairwiseMedianTest(formula = NULL, data = NULL, x = NULL, g = NULL,
digits = 4, method = "fdr", ...)

Arguments
formula  A formula indicating the measurement variable and the grouping variable. e.g. y ~ group.
data  The data frame to use.
x  The response variable as a vector.
g  The grouping variable as a vector.
digits  The number of significant digits to round output.
method  The p-value adjustment method to use for multiple tests. See stats::p.adjust.
...  Additional arguments passed to codecoin::median_test.
pairwiseMedianTest

Details

The input should include either formula and data; or x, and g.

Mood’s median test compares medians among two or more groups. See http://rcompanion.org/handbook/F_09.html for further discussion of this test.

The pairwiseMedianTest function can be used as a post-hoc method following an omnibus Mood’s median test.

Value

A dataframe of the groups being compared, the p-values, and the adjusted p-values.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_09.html

See Also

pairwiseMedianMatrix

Examples

data(PoohPiglet)
PoohPiglet$Speaker = factor(PoohPiglet$Speaker, 
  levels = c("Pooh", "Tigger", "Piglet"))
PT = pairwiseMedianTest(Likert ~ Speaker, 
  data = PoohPiglet, 
  exact = NULL, 
  method = "fdr")

PT
cldList(comparison = PT$Comparison, 
  p.value = PT$p.adjust, 
  threshold = 0.05)
pairwiseModelAnova  Compare model objects with F test and likelihood ratio test

Description

Compares a series of models with pairwise F tests and likelihood ratio tests.

Usage

pairwiseModelAnova(fits, ...)

Arguments

fits  A series of model object names, separated by commas.
...  Other arguments passed to list.

Details

For comparisons to be valid, both models must have the same data, without transformations, use the same dependent variable, and be fit with the same method.
To be valid, models need to be nested.

Value

A list of: The calls of the models compared; a data frame of comparisons and F tests; and a data frame of comparisons and likelihood ratio tests.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

See Also

compareGLM, compareLM

Examples

### Compare among polynomial models
data(BrendonSmall)
BrendonSmall$Calories = as.numeric(BrendonSmall$Calories)

BrendonSmall$Calories2 = BrendonSmall$Calories * BrendonSmall$Calories
BrendonSmall$Calories3 = BrendonSmall$Calories * BrendonSmall$Calories * BrendonSmall$Calories
BrendonSmall$Calories4 = BrendonSmall$Calories * BrendonSmall$Calories * BrendonSmall$Calories * BrendonSmall$Calories
model.1 = lm(Sodium ~ Calories, data = BrendonSmall)
model.2 = lm(Sodium ~ Calories + Calories2, data = BrendonSmall)
pairwiseNominalIndependence

Pairwise tests of independence for nominal data

Description

Conducts pairwise tests for a 2-dimensional matrix, in which at least one dimension has more than two levels, as a post-hoc test. Conducts Fisher exact, Chi-square, or G-test.

Usage

pairwiseNominalIndependence(x, compare = "row", fisher = TRUE, gtest = TRUE, chisq = TRUE, method = "fdr", correct = "none", cramer = FALSE, digits = 3, ...)

Arguments

x A two-way contingency table. At least one dimension should have more than two levels.
compare If "row", treats the rows as the grouping variable. If "column", treats the columns as the grouping variable.
fisher If "TRUE", conducts fisher exact test.
gtest If "TRUE", conducts G-test.
chisq If "TRUE", conducts Chi-square test of association.
method The method to adjust multiple p-values. See stats::p.adjust.
correct The correction method to pass to DescTools::GTest.
cramer If "TRUE", includes and effect size, Cramer's V in the output.
digits The number of significant digits in the output.
... Additional arguments, passed to stats::fisher.test, DescTools::GTest, or stats::chisq.test.

Value

A data frame of comparisons, p-values, and adjusted p-values.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>
pairwiseNominalMatrix

References

http://rcompanion.org/handbook/H_04.html

See Also

pairwiseMcnemar, groupwiseCMH, nominalSymmetryTest, pairwiseNominalMatrix

Examples

### Independence test for a 4 x 2 matrix

data(Anderson)
fisher.test(Anderson)
Anderson = Anderson[(c("Heimlich", "Bloom", "Dougal", "Cobblestone")),]
PT = pairwiseNominalIndependence(Anderson,
   fisher = TRUE,
   gtest = FALSE,
   chisq = FALSE,
   cramer = TRUE)

PT
cldList(comparison = PT$Comparison,
   p.value = PT$p.adj.Fisher,
   threshold = 0.05)

Pairwise tests of independence for nominal data with matrix output

Description

Conducts pairwise tests for a 2-dimensional matrix, in which at least one dimension has more than two levels, as a post-hoc test. Conducts Fisher exact, Chi-square, or G-test.

Usage

pairwiseNominalMatrix(x, compare = "row", fisher = TRUE,
   gtest = FALSE, chisq = FALSE, method = "fdr", correct = "none",
   digits = 3, ...)

Arguments

x A two-way contingency table. At least one dimension should have more than two levels.

compare If "row", treats the rows as the grouping variable. If "column", treats the columns as the grouping variable.

fisher If "TRUE", conducts Fisher exact test.

gtest If "TRUE", conducts G-test.
pairwiseOrdinalIndependence

chisq      If "TRUE", conducts Chi-square test of association.
method     The method to adjust multiple p-values. See p.adjust.
correct    The correction method to pass to DescTools::GTest.
digits     The number of significant digits in the output.
...        Additional arguments, passed to stats::fisher.test, DescTools::GTest, or stats::chisq.test.

Value

A list consisting of: the test used, a matrix of unadjusted p-values, the p-value adjustment method used, and a matrix of adjusted p-values.

@seealso pairwiseMcnemar, groupwiseCMH, nominalSymmetryTest, pairwiseNominalIndependence

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/H_04.html

Examples

### Independence test for a 4 x 2 matrix

data(Anderson)
fisher.test(Anderson)
Anderson = Anderson[(c("Heimlich", "Bloom", "Dougal", "Cobblestone")),]
PT = pairwiseNominalMatrix(Anderson,
                           fisher = TRUE,
                           gtest = FALSE,
                           chisq = FALSE)$Adjusted
PT
library(multcompView)
multcompLetters(PT)

---

pairwiseOrdinalIndependence

Pairwise tests of independence for tables with one ordered nominal variable

Description

Conducts pairwise tests for a 2-dimensional table, in which one variable is ordered nominal and one variable is non-ordered nominal. The function relies on the coin package.
Usage  
pairwiseOrdinalIndependence(x, compare = "row", scores = NULL,  
method = "fdr", digits = 3, ...)  

Arguments  
x  A two-way contingency table. One dimension is ordered and one is non-ordered nominal.  
compare  If "row", treats the rows as the grouping variable. If "column", treats the columns as the grouping variable.  
scores  Optional vector to specify the spacing of the ordered variable.  
method  The method to adjust multiple p-values. See stats::p.adjust.  
digits  The number of significant digits in the output.  
...  Additional arguments, passed to stats::chisq.test.  

Value  
A data frame of comparisons, p-values, and adjusted p-values.  

Author(s)  
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>  

References  
http://rcompanion.org/handbook/H_09.html  

See Also  
pairwiseNominalIndependence  

Examples  
### Independence test for table with one ordered variable  
data(Breakfast)  
require(coin)  
chisq_test(Breakfast,  
scores = list("Breakfast" = c(-2, -1, 0, 1, 2)))  
PT = pairwiseOrdinalIndependence(Breakfast, compare = "row")  
PT  
cldList(comparison = PT$Comparison,  
p.value = PT$p.value,  
threshold = 0.05)  

### Similar to Kruskal-Wallis test for Likert data  
data(PoohPiglet)  
XT = xtabs(~ Speaker + Likert, data = PoohPiglet)  
XT  
require(coin)
pairwiseOrdinalPairedMatrix

[Defunct!] Pairwise two-sample ordinal regression for paired data with matrix output

Description
Defunct. Performs pairwise two-sample ordinal regression across groups for paired data with matrix output.

Usage
pairwiseOrdinalPairedMatrix(...)

Arguments
... Anything.

pairwiseOrdinalMatrix

[Defunct!] Pairwise two-sample ordinal regression with matrix output

Description
Defunct. Performs pairwise two-sample ordinal regression across groups.

Usage
pairwiseOrdinalMatrix(...)
pairwiseOrdinalPairedTest

[Defunct!] Pairwise two-sample ordinal regression for paired data

Description
Defunct. Performs pairwise two-sample ordinal regression across groups for paired data.

Usage
pairwiseOrdinalPairedTest(...)

Arguments
...

Anything.

pairwiseOrdinalTest

[Defunct!] Pairwise two-sample ordinal regression

Description
Defunct. Performs pairwise two-sample ordinal regression across groups.

Usage
pairwiseOrdinalTest(...)

Arguments
...

Anything.
pairwisePercentileTest

Pairwise permutation tests for percentiles

Description
Conducts pairwise permutation tests across groups for percentiles, medians, and proportion below a threshold value.

Usage
pairwisePercentileTest(formula = NULL, data = NULL, x = NULL, y = NULL, test = "median", tau = 0.5, type = 7, threshold = NA, comparison = "<", r = 1000, digits = 4, progress = "TRUE", method = "fdr")

Arguments
formula A formula indicating the response variable and the independent variable. e.g. y ~ group.
data The data frame to use.
x If no formula is given, the response variable for one group.
y The response variable for the other group.
test The statistic to compare between groups. Can be "median", "percentile", "iqr", "proportion", "mean", or "variance".
tau If "percentile" is chosen as the test, tau indicates the percentile to test. Expressed as a quantile. That is, 0.5 indicates a test for medians. 0.75 indicates a test for 75th percentiles.
type The type value passed to the quantile function.
threshold If "proportion" is chosen as the test, threshold indicates the value of the dependent variable to use as the threshold. For example, to test if there is a different in the proportion of observations below $10,000, threshold = 10000 would be used.
comparison If "proportion" is chosen as the test, comparison indicates the inequality to use. Options are "<", "<=", ">", ">=", or ":="
r The number of replicates in the permutation test.
digits The number of significant digits in the output.
progress If TRUE, prints a dot for every 1 percent of the progress while conducting the test.
method The p-value adjustment method to use for multiple tests. See stats::p.adjust.
pairwisePercentileTest

Details
The function conducts pairwise tests using the percentileTest function. The user can consult the documentation for that function for additional details.
The input should include either formula and data; or x, and y.

Value
A dataframe of the groups being compared, the p-values, and the adjusted p-values.

Note
The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References
http://rcompanion.org/handbook/F_15.html

See Also
percentileTest, groupwisePercentile

Examples
```r
## Not run:
data(BrendonSmall)
PT = pairwisePercentileTest(Sodium ~ Instructor,
data = BrendonSmall,
test = "percentile",
tau = 0.75)
PT
cldList(p.adjust ~ Comparison,
data = PT,
threshold = 0.05)
data(BrendonSmall)
PT = pairwisePercentileTest(Sodium ~ Instructor,
data = BrendonSmall,
test = "proportion",
threshold = 1300)
PT
cldList(p.adjust ~ Comparison,
data = PT,
threshold = 0.05)
## End(Not run)
```
pairwisePermutationMatrix

Pairwise two-sample permutation tests with matrix output

Description
Conducts pairwise two-sample permutation tests across groups.

Usage
pairwisePermutationMatrix(formula = NULL, data = NULL, x = NULL, g = NULL, method = "fdr", ...)

Arguments
- formula: A formula indicating the measurement variable and the grouping variable. e.g. y ~ group.
- data: The data frame to use.
- x: The response variable as a vector.
- g: The grouping variable as a vector.
- method: The p-value adjustment method to use for multiple tests. See stats::p.adjust.
- ...: Additional arguments passed to coin::independence_test.

Details
The input should include either formula and data; or x, and g.
Permutation tests are non-parametric tests that do not assume normally-distributed errors. See http://rcompanion.org/rcompanion/d_06a.html for further discussion of this test.
The pairwisePermutationTest function can be used as a post-hoc method following an omnibus permutation test analogous to a one-way analysis of variance. The matrix output can be converted to a compact letter display.

Value
A list consisting of: A matrix of p-values; the p-value adjustment method; a matrix of adjusted p-values.

Note
The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>
pairwisePermutationSymmetry

References

http://rcompanion.org/handbook/K_02.html

See Also

pairwisePermutationTest

Examples

data(PoohPiglet)
PoohPiglet$Speaker = factor(PoohPiglet$Speaker,
levels = c("Pooh", "Tigger", "Piglet"))
PT = pairwisePermutationMatrix(Likert ~ Speaker,
data = PoohPiglet,
method = "fdr")
PT
PT = PT$Adjusted
library(multcompView)
multcompLetters(PT,
compare="<",
threshold=0.05,
Letters=letters)

pairwisePermutationSymmetry

Pairwise two-sample permutation symmetry tests

Description

Conducts pairwise two-sample permutation tests of symmetry across groups.

Usage

pairwisePermutationSymmetry(formula = NULL, data = NULL, x = NULL,
g = NULL, b = NULL, method = "fdr", ...)

Arguments

formula A formula indicating the measurement variable and the grouping variable. e.g.
y ~ group | block.
data The data frame to use.
x The response variable as a vector.
g The grouping variable as a vector.
b The blocking variable as a vector.
method The p-value adjustment method to use for multiple tests. See stats::p.adjust.
... Additional arguments passed to coin::symmetry_test.
Details

The input should include either formula and data; or x, g, and b.

Permutation tests are non-parametric tests that do not assume normally-distributed errors. See http://rcompanion.org/rcompanion/d_06a.html for further discussion of this test.

The `pairwisePermutationSymmetry` function can be used as a post-hoc method following an omnibus permutation test analogous to a paired one-way analysis of variance.

Value

A dataframe of the groups being compared, the p-values, and the adjusted p-values.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable. The second variable on the right side is used for the blocking variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/K_03.html

See Also

`pairwisePermutationSymmetryMatrix`

Examples

data(BobBelcher)
BobBelcher$Instructor = factor( BobBelcher$Instructor, 
    levels = c("Linda Belcher", "Louise Belcher", 
              "Tina Belcher", "Bob Belcher", 
              "Gene Belcher"))
BobBelcher$Likert.f = factor(BobBelcher$Likert, ordered=TRUE)
PT = pairwisePermutationSymmetry(Likert.f ~ Instructor | Rater, 
    data = BobBelcher, 
    method = "fdr")
PT
clList(comparison = PT$Comparison, 
    p.value = PT$p.adjust, 
    threshold = 0.05)
pairwisePermutationSymmetryMatrix

Pairwise two-sample permutation symmetry tests with matrix output

Description

Conducts pairwise two-sample permutation tests for symmetry across groups.

Usage

pairwisePermutationSymmetryMatrix(formula = NULL, data = NULL, x = NULL, g = NULL, b = NULL, method = "fdr", ...)

Arguments

formula A formula indicating the measurement variable and the grouping variable. e.g. y ~ group.
data The data frame to use.
x The response variable as a vector.
g The grouping variable as a vector.
b The blocking variable as a vector.
method The p-value adjustment method to use for multiple tests. See stats::p.adjust.
... Additional arguments passed to coin::symmetry_test.

Details

The input should include either formula and data; or x, g, and b.

Permutation tests are non-parametric tests that do not assume normally-distributed errors. See http://rcompanion.org/rcompanion/d_06a.html for further discussion of this test.

The pairwisePermutationSymmetryMatrix function can be used as a post-hoc method following an omnibus permutation test analogous to a paired one-way analysis of variance. The matrix output can be converted to a compact letter display.

Value

A list consisting of: A matrix of p-values; the p-value adjustment method; a matrix of adjusted p-values.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable. The second variable on the right side is used for the blocking variable.
Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/K_03.html

See Also

pairwisePermutationSymmetry

Examples

data(BobBelcher)
BobBelcher$Instructor = factor(BobBelcher$Instructor,
   levels = c("Linda Belcher", "Louise Belcher",
   "Tina Belcher", "Bob Belcher",
   "Gene Belcher"))
BobBelcher$Likert.f = factor(BobBelcher$Likert, ordered=TRUE)
PT = pairwisePermutationSymmetryMatrix(Likert.f ~ Instructor | Rater,
   data = BobBelcher,
   method = "fdr")$Adjusted
PT
library(multcompView)
multcompLetters(PT,
   compare = "<",
   threshold=0.05,
   Letters=letters)
Arguments

- **formula**: A formula indicating the measurement variable and the grouping variable. e.g. `y ~ group`.
- **data**: The data frame to use.
- **x**: The response variable as a vector.
- **g**: The grouping variable as a vector.
- **method**: The p-value adjustment method to use for multiple tests. See `stats::p.adjust`.
- **...**: Additional arguments passed to `coin::independence_test`.

Details

The input should include either `formula` and `data`; or `x`, and `g`.

Permutation tests are non-parametric tests that do not assume normally-distributed errors. See http://rcompanion.org/rcompanion/d_06a.html for further discussion of this test.

The `pairwisePermutationTest` function can be used as a post-hoc method following an omnibus permutation test analogous to a one-way analysis of variance.

Value

A dataframe of the groups being compared, the p-values, and the adjusted p-values.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/K_02.html

See Also

`pairwisePermutationMatrix`

Examples

data(PoohPiglet)
PoohPiglet$Speaker = factor(PoohPiglet$Speaker, levels = c("Pooh", "Tigger", "Piglet"))

PT = pairwisePermutationTest(Likert ~ Speaker,
data = PoohPiglet,
method = "fdr")

PT
cldList(comparison = PT$Comparison,
pairwiseRobustMatrix

[Defunct!] Pairwise two-sample robust tests with matrix output

Description

Defunct. Performs pairwise two-sample robust tests across groups with matrix output.

Usage

pairwiseRobustMatrix(...)

Arguments

... Anything.

pairwiseRobustTest [Defunct!] Pairwise two-sample robust tests

Description

Defunct. Performs pairwise two-sample robust tests across groups.

Usage

pairwiseRobustTest(...)

Arguments

... Anything.
pairwiseSignMatrix  [Defunct!] Pairwise sign tests with matrix output

Description
Defunct. Performs pairwise sign tests.

Usage
pairwiseSignMatrix(...)

Arguments
...  Anything.

pairwiseSignTest  [Defunct!] Pairwise sign tests

Description
Defunct. Performs pairwise sign tests.

Usage
pairwiseSignTest(...)

Arguments
...  Anything.

Pennsylvania18  Votes for the Democratic candidate in Pennsylvania 18 in 2016 and 2018

Description
A two-by-two matrix with the proportion of votes for the Democratic candidate in two races, in 2016 and 2018. 2016 is the Presidential election with Hillary Clinton as the Democratic candidate. 2018 is a House of Representatives election with Conor Lamb. These data are for Pennsylvania’s 18th Congressional District.

Usage
Pennsylvania18
percentileTest

Format
An object of class matrix with 2 rows and 2 columns.

Source
http://rcompanion.org/handbook/H_10.html

percentileTest Test of percentiles by permutation test

Description
Conducts a permutation test to compare two groups for medians, percentiles, or proportion below a threshold value.

Usage
percentileTest(formula = NULL, data = NULL, x = NULL, y = NULL, test = "median", tau = 0.5, type = 7, threshold = NA, comparison = "<", r = 1000, digits = 4, progress = "TRUE")

Arguments

formula A formula indicating the response variable and the independent variable. e.g. y ~ group.
data The data frame to use.
x If no formula is given, the response variable for one group.
y The response variable for the other group.
test The statistic to compare between groups. Can be "median", "percentile", "iqr", "proportion", "mean", or "variance".
tau If "percentile" is chosen as the test, tau indicates the percentile to test. Expressed as a quantile. That is, 0.5 indicates a test for medians. 0.75 indicates a test for 75th percentiles.
type The type value passed to the quantile function.
threshold If "proportion" is chosen as the test, threshold indicates the value of the dependent variable to use as the threshold. For example, to test if there is a different in the proportion of observations below $10,000, threshold = 10000 would be used.
comparison If "proportion" is chosen as the test, comparison indicates the inequality to use. Options are "<", "<=", ",", ",>=", or "="
r The number of replicates in the permutation test.
digits The number of significant digits in the output.
progress If TRUE, prints a dot for every 1 percent of progress while conducting the test.
Details

The function will test for a difference in medians, percentiles, interquartile ranges, proportion of observations above or below some threshold value, means, or variances between two groups by permutation test.

The input should include either formula and data; or x and y.

The function removes cases with NA in any of the variables.

If the independent variable has more than two groups, only the first two levels of the factor variable will be used.

The p-value returned is a two-sided test.

Value

A list of three data frames with the data used, a summary for each group, and the p-value from the test.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the independent variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_15.html

Examples

data(BrendonSmall)
percentileTest(Sodium ~ Instructor,
data=BrendonSmall,
test="median")

percentileTest(Sodium ~ Instructor,
data=BrendonSmall,
test="percentile",
tau = 0.75)

percentileTest(Sodium ~ Instructor,
data=BrendonSmall,
test="proportion",
threshold = 1300)
plotDensityHistogram

*Histogram with a density curve*

**Description**

Produces a histogram for a vector of values and adds a density curve of the distribution.

**Usage**

```r
plotDensityHistogram(x, prob = FALSE, col = "gray", main = "", linecol = "black", lwd = 2, adjust = 1, bw = "nrd0", kernel = "gaussian", ...)
```

**Arguments**

- **x**  
  A vector of values.

- **prob**  
  If FALSE, then counts are displayed in the histogram. If TRUE, then the density is shown.

- **col**  
  The color of the histogram bars.

- **main**  
  The title displayed for the plot.

- **linecol**  
  The color of the line in the plot.

- **lwd**  
  The width of the line in the plot.

- **adjust**  
  Passed to density. A lower value makes the density plot smoother.

- **bw**  
  Passed to density.

- **kernel**  
  Passed to density.

- **...**  
  Other arguments passed to hist.

**Details**

The function relies on the hist function. The density curve relies on the density function.

**Value**

Produces a plot. Returns nothing.

**Author(s)**

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

**References**

http://rcompanion.org/handbook/C_04.html
plotNormalDensity

See Also

plotNormalHistogram plotNormalDensity

Examples

### Plot of residuals from a model fit with lm

data(Catbus)
model = lm(Steps ~ Sex + Teacher,
data = Catbus)
plotDensityHistogram(residuals(model))

plotNormalDensity

Density plot with a normal curve

Description

Produces a density plot for a vector of values and adds a normal curve with the same mean and standard deviation. The plot can be used to quickly compare the distribution of data to a normal distribution.

Usage

plotNormalDensity(x, col1 = "white", col2 = "gray", col3 = "blue",
border = NA, main = "", lwd = 2, length = 1000, adjust = 1,
bw = "nrd0", kernel = "gaussian", ...)

Arguments

x A vector of values.
col1 The color of the density plot. Usually not visible.
col2 The color of the density polygon.
col3 The color of the normal line.
border The color of the border around the density polygon.
main The title displayed for the plot.
lwd The width of the line in the plot.
length The number of points in the line in the plot.
adjust Passed to density. A lower value makes the density plot smoother.
bw Passed to density.
kernelpassed to density.
... Other arguments passed to plot.
Details

The function plots a polygon based on the `density` function. The normal curve has the same mean and standard deviation as the values in the vector.

Value

Produces a plot. Returns nothing.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/I_01.html

See Also

plotNormalHistogram, plotDensityHistogram

Examples

```r
### Plot of residuals from a model fit with lm
data(Catbus)
model = lm(Steps ~ Sex + Teacher,
          data = Catbus)
plotNormalDensity(residuals(model))
```

```
plotNormalHistogram
Histogram with a normal curve
```

Description

Produces a histogram for a vector of values and adds a normal curve with the same mean and standard deviation. The plot can be used to quickly compare the distribution of data to a normal distribution.

Usage

```r
plotNormalHistogram(x, prob = FALSE, col = "gray", main = ",
                   linecol = "blue", lwd = 2, length = 1000, ...)
```
plotNormalHistogram

Arguments

- `x` A vector of values.
- `prob` If `FALSE`, then counts are displayed in the histogram. If `TRUE`, then the density is shown.
- `col` The color of the histogram bars.
- `main` The title displayed for the plot.
- `linecol` The color of the line in the plot.
- `lwd` The width of the line in the plot.
- `length` The number of points in the line in the plot.
- `...` Other arguments passed to `hist`.

Details

The function relies on the `hist` function. The normal curve has the same mean and standard deviation as the values in the vector.

Value

Produces a plot. Returns nothing.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/I_01.html

See Also

plotNormalDensity, plotDensityHistogram

Examples

```r
### Plot of residuals from a model fit with lm
data(Catbus)
model = lm(Steps ~ Sex + Teacher,
           data = Catbus)
plotNormalHistogram(residuals(model))
```
Description

Plots the best fit line for a model with one y variable and one x variable, or with one y variable and polynomial x variables.

Usage

\[
\text{plotPredy(} \text{data, x, y, model, order = 1, x2 = NULL, x3 = NULL, x4 = NULL, x5 = NULL, pch = 16, xlab = "X", ylab = "Y", length = 1000, lty = 1, lwd = 2, col = "blue", type = NULL, \ldots)\]

Arguments

- **data**: The name of the data frame.
- **x**: The name of the x variable.
- **y**: The name of the y variable.
- **model**: The name of the model object.
- **order**: If plotting a polynomial function, the order of the polynomial. Otherwise can be left as 1.
- **x2**: If applicable, the name of the second order polynomial x variable.
- **x3**: If applicable, the name of the third order polynomial x variable.
- **x4**: If applicable, the name of the fourth order polynomial x variable.
- **x5**: If applicable, the name of the fifth order polynomial x variable.
- **pch**: The shape of the plotted data points.
- **xlab**: The label for the x-axis.
- **ylab**: The label for the y-axis.
- **length**: The number of points used to draw the line.
- **lty**: The style of the plotted line.
- **lwd**: The width of the plotted line.
- **col**: The col of the plotted line.
- **type**: Passed to predict. Required for certain models.
- **\ldots**: Other arguments passed to plot.

Details

Any model for which predict() is defined can be used.
Value

Produces a plot. Returns nothing.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/I_10.html

Examples

```r
### Plot of linear model fit with lm
data(BrendonSmall)
model = lm(Weight ~ Calories, data = BrendonSmall)
plotPredy(data = BrendonSmall,
y = Weight,
x = Calories,
model = model,
xlab = "Calories per day",
ylab = "Weight in kilograms")

### Plot of polynomial model fit with lm
BrendonSmall$Calories2 = BrendonSmall$Calories * BrendonSmall$Calories
model = lm(Sodium ~ Calories + Calories2, data = BrendonSmall)
plotPredy(data = BrendonSmall,
y = Sodium,
x = Calories,
x2 = Calories2,
model = model,
order = 2,
xlab = "Calories per day",
ylab = "Sodium intake per day")

### Plot of quadratic plateau model fit with nls
quadplat = function(x, a, b, clx) {
    ifelse(x < clx, a + b * x + (-0.5*b/clx) * x * x,
           a + b * clx + (-0.5*b/clx) * clx * clx)
} 
model = nls(Sodium ~ quadplat(Calories, a, b, clx),
data = BrendonSmall,
start = list(a = 519,
b = 0.359,
clx = 2304))
plotPredy(data = BrendonSmall,
y = Sodium,
x = Calories,
model = model,
xlab = "Calories per day",
ylab = "Sodium intake per day")
```
### Logistic regression example requires type option

```r
data(BullyHill)
Trials = cbind(BullyHill$Pass, BullyHill$Fail)
model.log = glm(Trials ~ Grade, data = BullyHill,
    family = binomial(link="logit"))
plotPredy(data = BullyHill,
    y = Percent,
    x = Grade,
    model = model.log,
    type = "response",
    xlab = "Grade",
    ylab = "Proportion passing")
```

---

**Pooh**

*Hypothetical data for paired ratings of Pooh Bear*

**Description**

A data frame of Likert responses for instructor Pooh Bear for each of 10 respondents, paired before and after. Hypothetical data.

**Usage**

```r
Pooh
```

**Format**

An object of class `data.frame` with 20 rows and 4 columns.

**Source**

http://rcompanion.org/handbook/F_06.html

---

**PoohPiglet**

*Hypothetical data for ratings of Pooh, Piglet, and Tigger*

**Description**

A data frame of Likert responses for instructors Pooh Bear, Piglet, and Tigger. Hypothetical data.

**Usage**

```r
PoohPiglet
```
**Format**

An object of class `data.frame` with 30 rows and 2 columns.

**Source**

http://rcompanion.org/handbook/F_08.html

---

**rcompanion**    **Functions to Support Extension Education Program Evaluation**

---

**Description**

Functions and datasets to support Summary and Analysis of Extension Program Evaluation in R and An R Companion for the Handbook of Biological Statistics.

**Useful functions**

There are several functions that provide summary statistics for grouped data. These function titles tend to start with "`groupwise`". They provide means, medians, geometric means, and Huber M-estimators for groups, along with confidence intervals by traditional methods and bootstrap.

Functions to produce effect size statistics, some with bootstrapped confidence intervals, include those for Cramer’s V, Cohen’s g and odds ratio for paired tables, Cohen’s h, Cohen’s w, Vargha and Delaney’s A, Cliff’s delta, r for one-sample, two-sample, and paired Wilcoxon and Mann-Whitney tests, epsilon-squared, and Freeman’s theta.

There are also functions that are useful for comparing models. `compareLM`, `compareGLM`, and `pairwiseModelAnova`. These use goodness-of-fit measures like AIC, BIC, and BICc, or likelihood ratio tests. The accuracy function reports statistics for models including minimum maximum accuracy, MAPE, RMSE, Efron’s pseudo r-squared, and coefficient of variation.

Functions for nominal data include post-hoc tests for Cochran-Mantel-Haenszel test (`groupwiseCMH`), for McNemar-Bowker test (`pairwiseMcnemar`), and for tests of association like Chi-square, Fisher exact, and G-test (`pairwiseNominalIndependence`).

There are a few useful plotting functions, including `plotNormalHistogram` that plots a histogram of values and overlays a normal curve, and `plotPredy` which plots of line for predicted values for a bivariate model. Other plotting functions include producing density plots.

The function `nagelkerke` provides pseudo R-squared values for a variety of model types, as well as a likelihood ratio test for the model as a whole.

A function close to my heart is (`cateNelson`), which performs Cate-Nelson analysis for bivariate data.

**Vignettes and examples**

The functions in this package are used in "Extension Education Program Evaluation in R" which is available at http://rcompanion.org/handbook/ and "An R Companion for the Handbook of Biological Statistics" which is available at http://rcompanion.org/rcompanion/.

The documentation for each function includes an example as well.
Version notes

Version 2.0 is not entirely back-compatible as several functions have been removed. These include some of the pairwise methods that can be replaced with better methods. Also, some functions have been removed or modified in order to import fewer packages.

Removed packages are indicated with ‘Defunct’ in their titles.

---

| Religion | Hypothetical data for change in religion after a caucusing event |

Description

A matrix of paired counts for religion of people before and after an event. Hypothetical data.

Usage

Religion

Format

An object of class matrix with 4 rows and 4 columns.

Source

http://rcompanion.org/handbook/H_05.html

---

scheirerRayHare  Scheirer Ray Hare test

Description

Conducts Scheirer Ray Hare test.

Usage

scheirerRayHare(formula = NULL, data = NULL, y = NULL, x1 = NULL, x2 = NULL, tie.correct = TRUE, ss = TRUE, verbose = TRUE)
Arguments

- **formula**: A formula indicating the response variable and two independent variables. e.g. `y ~ x1 + x2`.
- **data**: The data frame to use.
- **y**: If no formula is given, the response variable.
- **x1**: The first independent variable.
- **x2**: The second independent variable.
- **tie.correct**: If "TRUE", applies a correction for ties in the response variable.
- **ss**: If "TRUE", includes the sums of squares in the output.
- **verbose**: If "TRUE", outputs statistics used in the analysis by direct print.

Details

The Scheirer Ray Hare test is a nonparametric test used for a two-way factorial experiment. It is described by Sokal and Rohlf (1995). At the time of writing, it does not appear to be a common or well-regarded test. It is sometimes recommended that the design should be balanced, and that there should be at least five observations for each cell in the interaction.

The input should include either `formula` and `data`; or `y`, `x1`, and `x2`.

The function removes cases with NA in any of the variables.

Value

A data frame of results similar to an anova table. Output from the `verbose` option is printed directly and not returned with the data frame.

Note

The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the first independent variable. The second variable on the right side is used for the second independent variable.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References


### Examples

```r
## Example from Sokal and Rohlf, 1995.
Value = c(709,679,699,657,594,677,592,538,476,508,505,539)
Sex = c(rep("Male",3), rep("Female",3), rep("Male",3), rep("Female",3))
Fat = c(rep("Fresh", 6), rep("Rancid", 6))
Sokal = data.frame(Value, Sex, Fat)
scheirerRayHare(Value ~ Sex + Fat, data=Sokal)
```

---

**spearmanRho**  
*Spearman's rho, Kendall's tau, Pearson's r*

### Description

Calculates Spearman's rho, Kendall's tau, or Pearson's r with confidence intervals by bootstrap.

### Usage

```r
spearmanRho(formula = NULL, data = NULL, x = NULL, y = NULL, 
method = "spearman", ci = FALSE, conf = 0.95, type = "perc", 
R = 1000, histogram = FALSE, digits = 3, ...)
```

### Arguments

- **formula**: A formula indicating the two paired variables, e.g. `~ x + y`. The variables should be vectors of the same length.
- **data**: The data frame to use.
- **x**: If no formula is given, the values for one variable.
- **y**: The values for the other variable.
- **method**: One of "spearman", "kendall", or "pearson". Passed to `cor`.
- **ci**: If TRUE, returns confidence intervals by bootstrap. May be slow.
- **conf**: The level for the confidence interval.
- **type**: The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to `boot.ci`.
- **R**: The number of replications to use for bootstrap.
- **histogram**: If TRUE, produces a histogram of bootstrapped values.
- **digits**: The number of significant digits in the output.
- **...**: Additional arguments passed to the `cor` function.
Details

This function is a wrapper for \texttt{stats::cor} with the addition of confidence intervals.

The input should include either \texttt{formula} and \texttt{data}; or \texttt{x}, and \texttt{y}.

Currently, the function makes no provisions for \texttt{NA} values in the data. It is recommended that \texttt{NA}s be removed beforehand.

When the returned statistic is close to 0 or close to 1, or with small sample size, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, \( \rho \), \( \tau \), or \( r \). Or a small data frame consisting of \( \rho \), \( \tau \), or \( r \), and the lower and upper confidence limits.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/I_10.html

Examples

data(Catbus)
spearmanRho(~Steps + Rating, data=Catbus)

Description

Conducts Tukey’s Ladder of Powers on a vector of values to produce a more-normally distributed vector of values.

Usage

\begin{verbatim}
transformTukey(x, start = -10, end = 10, int = 0.025,
    plotit = TRUE, verbose = FALSE, quiet = FALSE, statistic = 1,
    returnLambda = FALSE)
\end{verbatim}
transformTukey

Arguments

- **x**: A vector of values.
- **start**: The starting value of lambda to try.
- **end**: The ending value of lambda to try.
- **int**: The interval between lambda values to try.
- **plotit**: If TRUE, produces plots of Shapiro-Wilks W or Anderson-Darling A vs. lambda, a histogram of transformed values, and a quantile-quantile plot of transformed values.
- **verbose**: If TRUE, prints extra output for Shapiro-Wilks W or Anderson-Darling A vs. lambda.
- **quiet**: If TRUE, doesn’t print any output to the screen.
- **statistic**: If 1, uses Shapiro-Wilks test. If 2, uses Anderson-Darling test.
- **returnLambda**: If TRUE, returns only the lambda value, not the vector of transformed values.

Details

The function simply loops through lambda values from `start` to `end` at an interval of `int`.

The function then chooses the lambda which maximizes the Shapiro-Wilks W statistic or minimizes the Anderson-Darling A statistic.

It may be beneficial to add a constant to the input vector so that all values are positive. For left-skewed data, a (Constant - X) transformation may be helpful. Large values may need to be scaled.

Value

The transformed vector of values. The chosen lambda value is printed directly.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/I_12.html

Examples

```r
### Log-normal distribution example
Conc = rlnorm(100)
Conc.trans = transformTukey(Conc)
```
VDA is an effect size statistic appropriate in cases where a Wilcoxon-Mann-Whitney test might be used. It ranges from 0 to 1, with 0.5 indicating stochastic equality, and 1 indicating that the first group dominates the second.

The function calculates VDA from the "W" U statistic from the `wilcox.test` function. Specifically, VDA = U / (n1 * n2).

The input should include either `formula` and `data`; or `x`, and `y`. If there are more than two groups, only the first two groups are used.

Currently, the function makes no provisions for `NA` values in the data. It is recommended that NAs be removed beforehand.

When the data in the first group are greater than in the second group, vda is greater than 0.5. When the data in the second group are greater than in the first group, vda is less than 0.5. Be cautious with
this interpretation, as R will alphabetize groups in the formula interface if the grouping variable is not already a factor.

When VDA is close to 0 or close to 1, or with small sample size, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value
A single statistic, VDA. Or a small data frame consisting of VDA, and the lower and upper confidence limits.

Note
The parsing of the formula is simplistic. The first variable on the left side is used as the measurement variable. The first variable on the right side is used for the grouping variable.

Author(s)
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References
http://rcompanion.org/handbook/F_04.html

See Also
ciffDelta, multiVDA

Examples

data(Catbus)
  vda(Steps ~ Sex, data=Catbus)

wilcoxonOneSampleR  r effect size for Wilcoxon one-sample signed-rank test

Description
Calculates r effect size for a Wilcoxon one-sample signed-rank test; confidence intervals by bootstrap.

Usage
wilcoxonOneSampleR(x, mu = NULL, ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3, ...)

Arguments

- `x`: A vector of observations of an ordinal variable.
- `mu`: The value to compare `x` to, as in `wilcox.test`.
- `ci`: If TRUE, returns confidence intervals by bootstrap. May be slow.
- `conf`: The level for the confidence interval.
- `type`: The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to `boot.ci`.
- `R`: The number of replications to use for bootstrap.
- `histogram`: If TRUE, produces a histogram of bootstrapped values.
- `digits`: The number of significant digits in the output.
- `...`: Additional arguments passed to the `wilcoxsign_test` function.

Details

A Z value is extracted from the `wilcoxsign_test` function in the coin package. r is calculated as Z divided by square root of the number of observations.

The calculated statistic is equivalent to the statistic returned by the `wilcoxPairedR` function with one group equal to a vector of `mu`. The author knows of no reference for this technique.

Currently, the function makes no provisions for `NA` values in the data. It is recommended that `NAs` be removed beforehand.

When the data are greater than `mu`, r is positive. When the data are less than `mu`, r is negative.

When r is close to extremes, or with small counts in some cells, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, r. Or a small data frame consisting of r, and the lower and upper confidence limits.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_02.html

Examples

data(Pooh)
Data = Pooh[Pooh$Time==2,]
wilcox.test(Data$Likert, mu=3, exact=FALSE)
wilcoxonOneSampleR(x = Data$Likert, mu=3)
wilcoxonPairedR  r effect size for Wilcoxon two-sample paired signed-rank test

Description

Calculates r effect size for a Wilcoxon two-sample paired signed-rank test; confidence intervals by bootstrap.

Usage

wilcoxonPairedR(x, g = NULL, ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3, ...)

Arguments

x
A vector of observations of an ordinal variable.

g
The vector of observations for the grouping, nominal variable. Only the first two levels of the nominal variable are used. The data must be ordered so that the first observation of the of the first group is paired with the first observation of the second group.

ci
If TRUE, returns confidence intervals by bootstrap. May be slow.

conf
The level for the confidence interval.

type
The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.

R
The number of replications to use for bootstrap.

histogram
If TRUE, produces a histogram of bootstrapped values.

digits
The number of significant digits in the output.

...
Additional arguments passed to the wilcoxsion_test function.

Details

A Z value is extracted from the wilcoxsion_test function in the coin package. r is calculated as Z divided by square root of the number of observations in one group. This results in a statistic that ranges from 0 to 1.

Currently, the function makes no provisions for NA values in the data. It is recommended that NAs be removed beforehand.

When the data in the first group are greater than in the second group, r is positive. When the data in the second group are greater than in the first group, r is negative. Be cautious with this interpretation, as R will alphabetize groups if g is not already a factor.

When r is close to extremes, or with small counts in some cells, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, r. Or a small data frame consisting of r, and the lower and upper confidence limits.
wilcoxonR

Author(s)
Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References
http://rcompanion.org/handbook/F_06.html

Examples
data(Pooh)
wilcox.test(Likert ~ Time, data=Pooh, paired=TRUE, exact=FALSE)
wilcoxonPairedR(x = Pooh$Likert, g = Pooh$Time)

wilcoxonR  r effect size for Wilcoxon two-sample rank-sum test

Description
Calculates r effect size for Mann-Whitney, two-sample rank-sum test, or a table with an ordinal variable and a nominal variable with two levels; confidence intervals by bootstrap.

Usage
wilcoxonR(x, g = NULL, group = "row", ci = FALSE, conf = 0.95, type = "perc", R = 1000, histogram = FALSE, digits = 3, ...)

Arguments
x Either a two-way table or a two-way matrix. Can also be a vector of observations of an ordinal variable.
g If x is a vector, g is the vector of observations for the grouping, nominal variable. Only the first two levels of the nominal variable are used.
group If x is a table or matrix, group indicates whether the "row" or the "column" variable is the nominal, grouping variable.
ci If TRUE, returns confidence intervals by bootstrap. May be slow.
conf The level for the confidence interval.
type The type of confidence interval to use. Can be any of "norm", "basic", "perc", or "bca". Passed to boot.ci.
R The number of replications to use for bootstrap.
histogram If TRUE, produces a histogram of bootstrapped values.
digits The number of significant digits in the output.
... Additional arguments passed to the wilcox_test function.
Details

A Z value is extracted from the `wilcox_test` function in the coin package. \( r \) is calculated as \( Z \) divided by square root of the total observations.

Currently, the function makes no provisions for \( \text{NA} \) values in the data. It is recommended that \( \text{NAs} \) be removed beforehand.

When the data in the first group are greater than in the second group, \( r \) is positive. When the data in the second group are greater than in the first group, \( r \) is negative. Be cautious with this interpretation, as R will alphabetize groups if \( g \) is not already a factor.

When \( r \) is close to extremes, or with small counts in some cells, the confidence intervals determined by this method may not be reliable, or the procedure may fail.

Value

A single statistic, \( r \). Or a small data frame consisting of \( r \), and the lower and upper confidence limits.

Author(s)

Salvatore Mangiafico, <mangiafico@njaes.rutgers.edu>

References

http://rcompanion.org/handbook/F_04.html

See Also

freemanTheta

Examples

data(Breakfast)
Table = Breakfast[1:2,]
library(coin)
chisq_test(Table, scores = list("Breakfast" = c(-2, -1, 0, 1, 2)))
wilcoxonR(Table)

data(Catbus)
wilcox.test(Steps ~ Sex, data = Catbus)
wilcoxonR(x = Catbus$Steps, g = Catbus$Sex)