# Package ‘s20x’

June 22, 2018

**Version** 3.1-27  
**Date** 2018-06-22  
**Title** Functions for University of Auckland Course STATS 201/208 Data Analysis  
**Author** Andrew Balemi, Dineika Chandra, James Curran, Brant Deppa, Mike Forster, Brendan McArdle, Ben Stevenson, Chris Wild  
**Maintainer** James Curran <j.curran@auckland.ac.nz>  
**Description** A set of functions used in teaching STATS 201/208 Data Analysis at the University of Auckland. The functions are designed to make parts of R more accessible to a large undergraduate population who are mostly not statistics majors.  
**Imports** stats, graphics  
**License** GPL-2 | file LICENSE  
**RooxygenNote** 6.0.1  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2018-06-22 05:04:32 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>airpass.df</td>
<td>3</td>
</tr>
<tr>
<td>apples.df</td>
<td>3</td>
</tr>
<tr>
<td>arousal.df</td>
<td>4</td>
</tr>
<tr>
<td>autocorPlot</td>
<td>4</td>
</tr>
<tr>
<td>beer.df</td>
<td>5</td>
</tr>
<tr>
<td>body.df</td>
<td>5</td>
</tr>
<tr>
<td>books.df</td>
<td>5</td>
</tr>
<tr>
<td>boxqq</td>
<td>6</td>
</tr>
<tr>
<td>bursary.df</td>
<td>7</td>
</tr>
<tr>
<td>butterfat.df</td>
<td>7</td>
</tr>
<tr>
<td>camplake.df</td>
<td>8</td>
</tr>
<tr>
<td>chalk.df</td>
<td>8</td>
</tr>
</tbody>
</table>
## R topics documented:

- `ciReg` [9]
- `computer.df` [9]
- `cooks20x` [10]
- `course.df` [11]
- `course2way.df` [11]
- `crossFactors` [12]
- `crosstabs` [13]
- `diamonds.df` [14]
- `eovcheck` [14]
- `estimateContrasts` [16]
- `fire.df` [17]
- `freq1way` [17]
- `fruitfly.df` [19]
- `getVersion` [19]
- `house.df` [20]
- `incomes.df` [20]
- `interactionPlots` [20]
- `lakemary.df` [22]
- `larain.df` [22]
- `layout20x` [23]
- `levene.test` [23]
- `mazda.df` [24]
- `mening.df` [24]
- `mergers.df` [25]
- `modcheck` [25]
- `mozart.df` [26]
- `multipleComp` [27]
- `nail.df` [27]
- `normcheck` [28]
- `onewayPlot` [30]
- `oysters.df` [31]
- `pairs20x` [32]
- `peru.df` [32]
- `predict20x` [33]
- `predictCount` [34]
- `proplsd.new` [35]
- `rain.df` [36]
- `residPlot` [36]
- `rowdistr` [37]
- `rr` [38]
- `s0x` [38]
- `seeds.df` [39]
- `sheep.df` [39]
- `skewness` [39]
- `skulls.df` [40]
- `soyabean.df` [41]
- `stripqq` [41]
- `summary1way` [42]
airpass.df

*International Airline Passengers*

**Description**

Number of international airline passengers (in thousands) recorded monthly from January 1949 to December 1960.

**Format**

A time series with 144 observations.

apples.df

*Apples Data*

**Description**

These data come from a classic long-term experiment conducted at the East Malling Research Station, Kent, which is the centre for research into apple growing in the U.K. Commercial apple trees consist of two parts grafted together. The lowest part, the rootstock, largely determines the size of the tree, while the upper part (the scion) determines the fruit characteristics. Rootstocks propagated by cuttings (i.e. asexually produced) were once thought to result in smaller trees than those propagated from seeds (i.e. sexually produced). This hypothesis was re-examined in an experiment begun in 1918. Several trees of each type of 16 types of rootstock were planted, all trees having the same scion. Rootstocks I-IX were asexually produced, while X-XVI were sexually produced. In the winter of 1933-4 a number of trees were removed to make room for more, and the data presented here consists of the above ground weights of 104 trees felled in this period. No trees of types VIII, XI or XIV were felled. The description is from Lee (*Lee, A.J. Data analysis. An introduction based on R. University of Auckland 1994*). The data are from Andrews and Herzberg (1985).

**Format**

The data consist of a data frame with 104 observations on 3 variables.

```
[,1] Rootstock factor levels (I, II, III, IV, IX, V, VI, VII, X, XII, XIII, XV, XVI)
[,2] Weight integer .
[,3] Propagated factor levels (cutting, seed)
```
**arousal.df**

*Changes in Pupil Size with Emotional Arousal*

**Description**

Data from an experiment to measure the effect of different images on emotional arousal, by measuring changes in pupil diameter. The experiment used 20 males and 20 females. Images included a nude man, nude woman, infant, and a landscape.

**Format**

A data frame with 160 observations on 3 variables.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>arousal</td>
<td>numeric</td>
</tr>
<tr>
<td>[2]</td>
<td>gender</td>
<td>factor</td>
</tr>
<tr>
<td>[3]</td>
<td>picture</td>
<td>factor</td>
</tr>
</tbody>
</table>

---

**autocorPlot**

*Autocorrelation Plot*

**Description**

Plots current vs lagged residuals along with quadrants dividing these residuals about the value zero.

**Usage**

`autocorPlot(fit, main = "Current vs Lagged residuals", ...)`

**Arguments**

- `fit` output from the function `lm()`.
- `main` the plot title.
- `...` extra parameters to be passed to the `plot` function.

**Value**

Plots current vs lagged residuals along with quadrants dividing these residuals about the value zero.

**Note**

`autocor.plot` is deprecated inline with our new policy of removing periods from function names.
Examples

```r
data(airpass.df)
time = 1:144
airpass.fit = lm(passengers ~ time, data = airpass.df)
antocorPlot(airpass.fit)
```

beer.df  

US Beer Production

Description

Monthly United States beer production figures (in millions of 31-gallon barrels) for the period July 1970 to June 1978.

Format

A time series with 96 observations.

body.df  

Body Image and Ethnicity

Description

Data collected to examine how women from various ethnic groups rate their body image. All subjects were slightly underweight for their body size.

Format

A data frame with 246 observations on 8 variables.

<table>
<thead>
<tr>
<th>[,1]</th>
<th>ethnicity factor</th>
<th>Subject’s ethnicity (Asian, Europn, Maori, Pacific)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[,2]</td>
<td>married</td>
<td>.</td>
</tr>
<tr>
<td>[,3]</td>
<td>bodyim factor</td>
<td>Subject’s rating of themself (slight.uw, right, slight.ow, mod.ow, very.ow)</td>
</tr>
<tr>
<td>[,4]</td>
<td>sm.ever</td>
<td>.</td>
</tr>
<tr>
<td>[,5]</td>
<td>weight</td>
<td>.</td>
</tr>
<tr>
<td>[,7]</td>
<td>age</td>
<td>.</td>
</tr>
<tr>
<td>[,8]</td>
<td>stressgp</td>
<td>.</td>
</tr>
</tbody>
</table>

books.df  

Books Data
Description

This data consists of 50 sentence lengths from each of 8 books. The books “Disclosure” and “Rising Sun” were written by Michael Crichton, whilst the others “Four Past Midnight”, “The Dark Half”, “Eye of the Dragon”, “The Shining”, “The Stand” and “The Tommy-Knockers” were written by Stephen King. The pages and sentences were chosen using a multistage design where the pages were selected at random, and then sentences within each page were selected at random. These data were collected by James Curran.

Format

The data frame consists of 400 observations on 2 variables.

| [,1] length | integer |

Description

Draws boxplots and normal quantile-quantile plots of x for each value of the grouping variable g

Usage

```r
boxqq(formula, ...) #\texttt{S3 method for class \textquoteleft formula\textquoteleft}
boxqq(formula, data = NULL, ...)
```

Arguments

- `formula`: A symbolic specification of the form `x ~ g` can be given, indicating the observations in the vector `x` are to be grouped according to the levels of the factor `g`. `NA`'s are allowed in the data.
- `...`: Arguments to be passed to methods, such as graphical parameters (see `par`).
- `data`: An optional data frame in which to evaluate the formula.

Value

Returns the plot.

Methods (by class)

- `formula`: Box plots and normal quantile-quantile plots
Note

This function is deprecated and will be removed in later versions of the package.

Examples

```r
## Zoo data
data(zoo.df)
boxqq(attendance ~ day.type, data = zoo.df)
```

### bursary.df

**Bursary Results for Auckland Secondary Schools**

**Description**

Data for the 2001 Bursary results for 75 secondary schools in the Auckland area. For each school the decile rating of the school is recorded along with the percentage of eligible students who gain a B Bursary or better.

**Format**

A data frame with 75 observations on 2 variables.

<table>
<thead>
<tr>
<th>[,1]</th>
<th>decile</th>
<th>numeric</th>
<th>Decile rating of the school</th>
</tr>
</thead>
<tbody>
<tr>
<td>[,2]</td>
<td>pass.rate</td>
<td>numeric</td>
<td>Percentage of eligible students who gained a 'B' Bursary or better</td>
</tr>
</tbody>
</table>

### butterfat.df

**Butterfat Data**

**Description**

This data gives the mean percentage of butterfat produced by different Canadian pure-bred diary cattle. There are five different breeds and two age groups, two years old and greater than five years old. For each combination of breed and age, there are measurements for 10 cows.

**Format**

A data frame with 100 observations on 3 variables.

<table>
<thead>
<tr>
<th>[,1]</th>
<th>Butterfat</th>
<th>numeric</th>
<th>Mean percentage of butterfat per cow</th>
</tr>
</thead>
<tbody>
<tr>
<td>[,2]</td>
<td>Breed</td>
<td>factor</td>
<td>Breed (ayrshire, canadian, guernesy, holst.fres, jersey)</td>
</tr>
<tr>
<td>[,3]</td>
<td>Age</td>
<td>factor</td>
<td>Age group (2yo, mature)</td>
</tr>
</tbody>
</table>
**Source**

A Handbook of Small Data Sets

**References**


---

**camplakeNdf**  
*Age and Length of Camp Lake Bluegills*

**Description**

66 bluegills were captured from Camp Lake, Minnesota. For each bluegill we have the length of the fish, its age in years and its age in scale radius.

**Format**

A data frame with 66 observations on 3 variables.

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Scale.Radius</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>numeric</td>
<td>numeric</td>
<td>numeric</td>
</tr>
<tr>
<td>2</td>
<td>Age</td>
<td>Scale Radius</td>
<td>Length</td>
</tr>
<tr>
<td>3</td>
<td>numeric</td>
<td>numeric</td>
<td>numeric</td>
</tr>
</tbody>
</table>

**chalkNdf**  
*Chalk Data*

**Description**

These data involve 11 laboratories and 2 brands of chalk. The laboratories tested the density of the chalk. The main interest was whether the different laboratories yielded the same density for the two different types of chalk.

**Format**

A data frame with 66 observations on 3 variables.

<table>
<thead>
<tr>
<th></th>
<th>Density</th>
<th>Lab</th>
<th>Chalk</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>numeric</td>
<td>integer</td>
<td>factor</td>
</tr>
<tr>
<td>2</td>
<td>Density</td>
<td>Lab</td>
<td>Chalk</td>
</tr>
<tr>
<td>3</td>
<td>numeric</td>
<td>integer</td>
<td>factor</td>
</tr>
</tbody>
</table>
ciReg  

Confidence Intervals for Regression models

Description
Calculates and prints the confidence intervals for the fitted model.

Usage
\[
\text{ciReg}(\text{fit}, \text{conf.level} = 0.95, \text{print.out} = \text{TRUE})
\]

Arguments
- \textit{fit} \hspace{1cm} an object of class \texttt{lm}, i.e. the output from \texttt{lm}.
- \textit{conf.level} \hspace{1cm} confidence level of the intervals.
- \textit{print.out} \hspace{1cm} if \text{TRUE}, print out the output on the screen.

Value
The function returns a two-column matrix containing the upper and lower endpoints of the intervals.

See Also
\texttt{lm, summary, anova}.

Examples
\begin{verbatim}
# Peruvian Indians data
data(peru.df)
fit=lm(BP ~ age + years + weight + height, data = peru.df)
ciReg(fit)
\end{verbatim}

computer.df  

Computer Questionnaire

Description
Data from a test to see if a questionnaire was properly designed. The questionnaire measures managers’ technical knowledge of computers. The test has 19 managers complete the questionnaire as well as rate their own technical expertise.

Format
A data frame with 19 observations on 2 variables.
Description

Draws a Cook’s distance plot.

Usage

cooksx(x, main = "Cook’s Distance plot", xlab = "observation number", ylab = "Cook’s distance", line = c(0.5, 1.2, 2), cex.labels = 1, axisOpt = list(xAxis = TRUE, yAxis = FALSE), ...)

Arguments

x an object of class lm, usually obtained by using the lm function.
main the plot title
xlab the x-axis title.
ylab the y-axis title.
line a vector of length 3 controlling the distances of the plot title, the x-axis title and the y-axis title from the axis in line units.
cex.labels a factor controlling the font size of the labels on suspected high influence points.
axisOpt a list of additional arguments that can be used to control the axes. At this point this list only contains one element xAxis which is logical. If xAxis == TRUE then the x-axis will be displayed, and clearly, if it is FALSE, then it will not.
... additional arguments are passed to plot and may provide some extra flexibility.

Value

Returns the plot and identifies the three highest Cook’s values

Examples

# Peruvian Indians data
data(peru.df)
peru.fit = lm(BP ~ age + years + I(years^2) + weight + height, data = peru.df)
cooksx(peru.fit)
### Description

Data from a summer school Stats 20x course. Each observation represents a single student.

### Format

A data frame with 146 observations on 15 variables.

| [1] | Grade          | factor | Final grade for the course (A, B, C, D) |
| [2] | Pass           | factor | Passed the course (No, Yes)             |
| [3] | Exam           | numeric| Mark in the final exam                  |
| [4] | Degree         | factor | Degree enrolled in (BA, BCom, BSc, Other) |
| [5] | Gender         | factor | Gender (Female, Male)                    |
| [6] | Attend         | factor | Regularly attended class (No, Yes)       |
| [7] | Assign         | numeric| Assignment mark                         |
| [8] | Test           | numeric| Test mark                                |
| [9] | B              | numeric| Mark for the short answer section of the exam |
| [10] | C              | numeric| Mark for the long answer section of the exam |
| [11] | MC             | numeric| Mark for the multiple choice section of the exam |
| [12] | Colour         | factor | Colour of the exam booklet (Blue, Green, Pink, Yellow) |
| [13] | Stage1         | factor | Stage one grade (A, B, C)                |
| [14] | Years.Since    | numeric| Number of years since doing Stage 1      |
| [15] | Repeat         | factor | Repeating the paper (No, Yes)            |

### Description

Data from a summer school Stats 20x course. Each observation represents a single student. It is of interest to see if there is a relationship between a student’s final examination mark and both their gender and whether they regularly attend lectures.

### Format

A data frame with 40 observations on 3 variables.

| [1] | Exam     | numeric | Final exam mark (out of 100) |
| [2] | Gender   | factor  | Gender (Female, Male)         |
| [2] | Attend   | factor  | Regularly attended or not (No, Yes) |
crossFactors  

_Crossed Factors_

**Description**

Computes a factor that has a level for each combination of the factors ‘fac1’ and ‘fac2’.

**Usage**

crossFactors(x, fac2 = NULL, ...)

```r
## Default S3 method:
crossFactors(x, fac2 = NULL, ...)
```

```r
## S3 method for class 'formula'
crossFactors(formula, fac2 = NULL, data = NULL, ...)
```

**Arguments**

- `x` the name of the first factor or a formula in the form `~ fac1 * fac2`
- `fac2` the name of the second factor - ignored if `x` is a formula.
- `...` Optional arguments
- `formula` a formula in the form `~ fac1 * fac2`
- `data` an optional data frame in which to evaluate the formula

**Value**

Returns a vector containing the factor which represents the interaction of the given factors.

**Methods (by class)**

- default: Crossed Factors
- formula: Crossed Factors

**Note**

This function actually returns a factor now instead of a character string, so coercion into a factor is no longer necessary.

**See Also**

`factor`. 
Examples

```r
## arousal data:
data(arousal.df)
gender.picture = crossFactors(arousal.df$gender, arousal.df$picture)
gender.picture

## arousal data:
data(arousal.df)
gender.picture = crossFactors(~ gender * picture, data = arousal.df)
gender.picture
```

---

**crosstabs**

*Crosstabulation of two variables*

Description

Produces a 2-way table of counts and the corresponding chi-square test of independence or homogeneity.

Usage

crosstabs(formula, data)

Arguments

- `formula` a symbolic description of the model to be fit: \(~\text{fac1} + \text{fac2}\); where \text{fac1} and \text{fac2} are vectors to be crosstabulated and treated internally as factors.
- `data` an optional data frame containing the variables in the model.

Value

An invisible list containing the following components:

- `row.props` a matrix of row proportions, i.e. cell counts divided by row marginals.
- `col.props` a matrix of column proportions, i.e. cell counts divided by column marginals.
- `Totals` a matrix containing the cell counts and the marginal totals.

Note

This function is deprecated and will be removed in future versions of the package.
Examples

```r
# body image data:
data(body.df)
crosstabs(~ ethnicity + married, body.df)
```

---

**diamonds.df**

*Prices and Weights of Diamonds*

---

**Description**

Prices of ladies’ diamond rings from a Singaporean retailer and the weight of their diamond stones.

**Format**

A data frame with 48 observations on 2 variables.

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>[.1]</td>
<td>price numeric</td>
<td>Price of ring (Singapore dollars)</td>
</tr>
<tr>
<td>[.2]</td>
<td>weight numeric</td>
<td>Weight of Diamond (carats)</td>
</tr>
</tbody>
</table>

---

**eovcheck**

*Testing for equality of variance plot*

---

**Description**

Plots the residuals versus the fitted (or predicted) values from a linear model. A horizontal line is drawn at y = 0, reflecting the fact that we expect the residuals to have a mean of zero. An optional lowess line is drawn if smoother is set to TRUE. This can be useful in determining whether a trend still exists in the residuals. An optional pair of lines is drawn at +/- 2 times the standard deviation of the residuals - which is estimated from the Residual Mean Square (Within group mean square = WGMS). This can be useful in highlighting potential outliers. If the model has one or two factors and no continuous variables, i.e. if it is a oneway or twoway ANOVA model, and levene = TRUE then the P-value from Levene’s test for equality variance is displayed in the top left hand corner, as long as the number of observations per group exceeds two.

**Usage**

```r
eovcheck(x, ...)  
```

```r
# S3 method for class 'formula'
eovcheck(x, data = NULL, xlab = "Fitted values",  
         ylab = "Residuals", col = NULL, smoother = FALSE, twosd = FALSE,  
         levene = FALSE, ...)
```
## S3 method for class 'lm'

```
eovcheck(x, smoother = FALSE, twosd = FALSE, levene = FALSE, ...)
```

### Arguments

- `x`:
  A linear model formula. Alternatively, a fitted `lm` object from a linear model.

- `data`:
  A data frame in which to evaluate the formula.

- `xlab`:
  A title for the x axis; see `title`.

- `ylab`:
  A title for the y axis; see `title`.

- `col`:
  A color for the lowess smoother line.

- `smoother`:
  If TRUE then a smoothed lowess line will be added to the plot.

- `twosd`:
  If TRUE then horizontal dotted lines will be drawn at +/-2sd.

- `levene`:
  If TRUE then the P-value from Levene’s test for equality of variance is displayed.

### Methods (by class)

- `formula`: Testing for equality of variance plot
- `lm`: Testing for equality of variance plot

### See Also

- `levene.test`

### Examples

```r
# one way ANOVA - oysters
data(oysters.df)
oyster.fit = lm(Oysters ~ Site, data = oysters.df)
eovcheck(oyster.fit)

# Same model as the previous example, but using eovcheck.formula
data(oysters.df)
eovcheck(Oysters ~ Site, data = oysters.df)

# A two-way model without interaction
data(soyabean.df)
soya.fit = lm(yield ~ planttime + cultivar, data = soyabean.df)
eovcheck(soya.fit)

# A two-way model with interaction
data(arousal.df)
arousal.fit = lm(arousal ~ gender * picture, data = arousal.df)
eovcheck(arousal.fit)
```
```r
# A regression model
data(peru.df)
peru.fit = lm(BP ~ height + weight + age + years, data = peru.df)
eovcheck(peru.fit)

# A time series model
data(airpass.df)
t = 1:144
month = factor(rep(1:12, 12))
airpass.df = data.frame(passengers = airpass.df$passengers, t = t, month = month)
airpass.fit = lm(log(passengers)[-1] - t[-1] + month[-1]
            + log(passengers)[-144], data = airpass.df)
eovcheck(airpass.fit)
```

### estimateContrasts

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculates and prints Tukey multiple confidence intervals for contrasts in one or two-way ANOVA.</td>
</tr>
</tbody>
</table>

### Usage

```r
estimateContrasts(contrast.matrix, fit, row = TRUE, alpha = 0.05, L = NULL)
```

### Arguments

- **contrast.matrix**: a matrix of contrast coefficients. Separate rows of the matrix contain the contrast coefficients for that particular contrast, and a column for level of the factor.
- **fit**: output from the `lm` function.
- **row**: if TRUE, and the ANOVA is two-way, then contrasts in the row effects are printed, otherwise contrasts in the column effects are printed. Ignored if the ANOVA is one-way.
- **alpha**: the nominal error rate for the multiple confidence intervals.
- **L**: number of contrasts. If NULL, L will be set to the number of rows in the contrast matrix, otherwise L will be as specified.

### Value

Returns a matrix whose rows correspond to the different contrasts being estimated and whose columns correspond to the point estimate of the contrast, the Tukey lower and upper limits of the confidence interval, the unadjusted p-value, the Tukey and Bonferroni p-values.
freq1way

Note

: This function is no longer exported as it should never be called by the user. It will ultimately be removed.

See Also

summary1way, summary2way, multipleComp

Examples

```r
## computer data:
data(computer.df)
computer.df = within(computer.df, {selfassess = factor(selfassess)})
computer.fit = lm(score ~ selfassess, data = computer.df)
contrast.matrix = matrix(c(-1/2, -1/2, 1), byrow = TRUE, nrow = 1, ncol = 3)
contrast.matrix
s20x:::estimateContrasts(contrast.matrix, computer.fit)
```

fire.df

<table>
<thead>
<tr>
<th>fire.df</th>
<th>Fire Damage and Distance from the Fire Station</th>
</tr>
</thead>
</table>

Description

House damage and distance from the fire station, of 15 house fires. Data collected by an insurance company for homes in a particular area.

Format

A data frame with 15 observations on 2 variables.

```
[1,] damage  numeric Damage ($000s)
[2,] distance numeric Distance from the fire station (miles)
```

freq1way

Analysis of 1-dimensional frequency tables

Description

If hypothprob is absent: prints confidence intervals for the true proportions, a Chi-square test for uniformity, confidence intervals for differences in proportions (no corrections for multiple comparisons and plots the proportions.
freq1way

Usage

freq1way(counts, hypothprob, conf.level = 0.95, addCIs = TRUE, digits = 4, arrowwid = 0.1, estimated = 0)

Arguments

counts A 1-way frequency table as produced by table.
hypothprob If present, a set of probabilities to test the cell counts against.
conf.level confidence level for the confidence interval, expressed as a decimal.
addCIs If true, adds confidence limits to plot of sample proportions.
digits used to control rounding of printout.
arrowwid controls width of arrowheads.
estimated default is 0. Subtracted from the df for the Chi-square test.

Details

If hypothprob is present: prints confidence intervals for the true proportions, a Chi-square test for the hypothesized probabilities, and plots the sample proportions (with attached confidence limits) alongside the corresponding hypothesized probabilities.

Value

An invisible list containing the following components:

CIs a matrix containing the confidence intervals.
exp a vector of the expected counts.
chi a vector of the components of Chi-square.

Note

These confidence intervals have been Bonferroni adjusted for multiple comparisons. This function has been deprecated and will be removed from future versions of the package.

Examples

```R
# Body image data:
data(body.df)
eth.table = with(body.df, table(ethnicity))
freq1way(eth.table)
freq1way(eth.table, hypothprob=c(0.2, 0.4, 0.3, 0.1))
```
Fruitfly Data

Description

This data gives fecundity for female fruitflies, Drosophila melanogaster. The fecundity is the number of eggs laid, per day, for the fruitfly’s first 14 days of life. There are three strains: A control group, NS, Nonselected Strain, as well as RS, a strain bred for resistance to DDT and SS, a strain bred for susceptibility to DDT. Each strain contains 25 measurements. It is of interest to compare the level of fecundity across strains.

Format

A data frame with 75 observations on 2 variables.

[,1] fecundity numeric Number of eggs laid, per day, per fruitfly
[,2] strain factor Strain of fruitfly (NS, RS, SS)

Source

A Handbook of Small Data Sets

References


getVersion

s20x package version number

Description

Returns the version number of the s20x package. This is useful if a student is has problems running commands and the maintainer needs to check the version number.

Usage

getVersion()

Examples

getVersion()
### house.df

**Sale and Advertised Prices of Houses**

**Description**

A random sample of 100 houses recently sold in Mt Eden, Auckland. For each house we have the advertised price and the actual sale price.

**Format**

A data frame with 100 observations on 2 variables.

```
[,1] advertised.price  numeric  Advertised price ($)
[,2] sell.price        numeric  Final sale price ($)
```

### incomes.df

**Mean Family Incomes**

**Description**

Random sample of 152 families giving their mean income ($000s). The sample was taken by an advertising agency over their area of operations.

### interactionPlots

**Interactions Plot for Two-way Analysis of Variance**

**Description**

Displays data with intervals for each combination of the two factors and shows the mean differences between levels of the first factor for each level of the second factor. Note that there should be more than one observation for each combination of factors.

**Usage**

```
interactionPlots(y, ...)
```

## Default S3 method:

```
interactionPlots(y, fac1 = NULL, fac2 = NULL,
    xlab = NULL, xlab2 = NULL, ylab = NULL, data.order = TRUE,
    exlim = 0.1, jitter = 0.02, conf.level = 0.95,
    interval.type = c("tukey", "hsd", "lsd", "ci"), pooled = TRUE,
    tick.length = 0.1, interval.distance = 0.2, col.width = 2/3,
    xlab.distance = 0.1, xlen = 1.5, ylen = 1, ...)```
interactionPlots

## S3 method for class 'formula'
interactionPlots(y, data = NULL, xlab = NULL,
    xlab2 = NULL, ylab = NULL, data.order = TRUE, exlim = 0.1,
    jitter = 0.02, conf.level = 0.95, interval.type = c("tukey", "hsd",
    "lsd", "ci"), pooled = TRUE, tick.length = 0.1, interval.distance = 0.2,
    col.width = 2/3, xlab.distance = 0.1, xlen = 1.5, ylen = 1, ...
)

**Arguments**

- **y**: either a formula of the form: `y~fac1+fac2` where `y` is the response and `fac1` and `fac2` are the two explanatory variables used as factors, or a single response vector.
- **...**: optional arguments.
- **fac1**: if `y` is a vector, then `fac1` contains the levels of factor 1 which correspond to the `y` value.
- **fac2**: if `y` is a vector, then `fac1` contains the levels of factor 2 which correspond to the `y` value.
- **xlab**: an optional label for the x-axis. If not specified the name of `fac1` will be used.
- **xlab2**: an optional label for the lines. If not specified the name of `fac2` will be used.
- **ylab**: An optional label for the y-axis. If not specified the name of `y` will be used.
- **data.order**: if TRUE the levels of `fac1` and `fac2` will be set to `unique(fac1)` and `unique(fac2)` respectively.
- **exlim**: provide extra limits.
- **jitter**: the amount of horizontal jitter to show in the plot. The actual jitter is determined as the function is called, and will likely be different each time the function is used.
- **conf.level**: confidence level of the intervals.
- **interval.type**: four options for intervals appearing on plot: 'tukey', 'hsd', 'lsd' or 'ci'.
- **pooled**: two options: pooled or unpooled standard deviation used for plotted intervals.
- **tick.length**: size of tick, in inches.
- **interval.distance**: distance, as a fraction of the column width, between the points and interval. This is in addition to the extra space allocated for the jitter.
- **col.width**: width of a factor 'column', as a fraction of the space between the centres of two columns.
- **xlab.distance**: distance of x-axis labels from bottom of plot, as a fraction of the overall height of the plot.
- **xlen, ylen**: character interspacing factor for horizontal (x) and vertical (y) spacing of the legend.
- **data**: an optional data frame containing the variables in the model.
Methods (by class)

- default: Interactions Plot for Two-way Analysis of Variance
- formula: Interactions Plot for Two-way Analysis of Variance

See Also

summary2way.

Examples

data(mtcars)
interactionPlots(wt ~ vs + gear, mtcars)

## note this usage is deprecated
data(mtcars)
with(mtcars, interactionPlots(wt, vs, gear))

lakemary.df

Ages and Lengths of Lake Mary Bluegills

Description

The ages and lengths of 78 bluegills captured from Lake Mary, Minnesota.

Format

A data frame with 78 observations on 2 variables.

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>numeric</td>
<td>Age of the fish (years)</td>
</tr>
<tr>
<td>2</td>
<td>numeric</td>
<td>Length at capture (mm)</td>
</tr>
</tbody>
</table>

larain.df

Los Angeles Rainfall

Description

Annual rainfall (in inches) for Los Angeles from 1908 to 1973.

Format

A time series with 66 observations.
Description

Allows an numRows by numCols matrix of plots to be displayed in a single plot. If the function is called with no arguments, then the plotting device layout will be reset to a single plot.

Usage

layout20x(numRows = 1, numCols = 1)

Arguments

numRows number of rows in plot array
numCols number of columns in plot array

Value

Function returns no value

Note

This function is deprecated. It will be removed in future versions of the package.

Examples

data(course.df)
layout20x(1, 2)
stripchart(course.df$Exam)
boxplot(course.df$Exam)

levene.test Levene test for the ANOVA Assumption

Description

Perform a Levene test for equal group variances in both one-way and two-way ANOVA. A table with the results is (normally) displayed.

Usage

levene.test(formula, data, digit = 5, show.table = TRUE)
Arguments

- **formula**: a symbolic description of the model to be fitted: response ~ fac1 + fac2.
- **data**: an optional data frame containing the variables in the model.
- **digit**: the number of decimal places to display.
- **show.table**: If this argument is FALSE then the output will be suppressed.

Value

A list with the following elements:

- **df**: degrees of freedom.
- **ss**: sum squares.
- **ms**: mean squares.
- **f.value**: F-statistic value.
- **p.value**: P-value.

See Also

- `crossFactors`, `anova`.

Examples

```r
##
data(computer.df)
levene.test(score ~ factor(selfassess), computer.df)
```

---

**mazda.df**  
*Year and Price of Mazda Cars*

---

Description

Prices and ages of 124 Mazda cars collected from the Melbourne Age newspaper in 1991.

Format

A data frame with 124 observations on 2 variables.

<table>
<thead>
<tr>
<th></th>
<th>price</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>numeric</td>
<td></td>
</tr>
<tr>
<td>[2]</td>
<td>numeric</td>
<td></td>
</tr>
</tbody>
</table>

---

**mening.df**  
*Monthly Notifications of Meningococcal Disease*
**mergers.df**

**Description**
This data shows the monthly number of notifications meningococcal disease in New Zealand from January 1990 to December 2001.

**Format**
A data frame with 144 observations on 3 variables: Month, Year and mening.

<table>
<thead>
<tr>
<th>mergers.df</th>
<th>Merger Days</th>
</tr>
</thead>
</table>

**Description**
A random selection of 38 consummated mergers from the USA, 1982, giving the number of days between the date the merger was announced and the date the merger became effective.

<table>
<thead>
<tr>
<th>modcheck</th>
<th>Model checking plots</th>
</tr>
</thead>
</table>

**Description**
Plots four model checking plots: an pred-res plot (residuals against predicted values), a Normal Quantile-Quantile (Q-Q) plot, a histogram of the residuals with a normal distribution super-imposed and a Cook’s Distance plot.

**Usage**

```r
modcheck(x, ...)  
## S3 method for class 'lm'
modcheck(x, plotOrder = 1:4, args = list(eovcheck =  
  list(smooth = FALSE, twosd = FALSE, levene = FALSE, ...), normcheck =  
  list(xlab = c("Theoretical Quantiles", ""), ylab = c("Sample Quantiles", ""),  
  main = c("", ""), col = "light blue", bootstrap = FALSE, B = 5, bpch = 3, bcol  
  = "lightgrey", shapiro.wilk = FALSE, whichPlot = 1:2, usePar = TRUE, ...),  
  cooks2ox = list(main = "Cook's Distance plot", xlab = "observation number",  
  ylab = "Cook's distance", line = c(0.5, 0.1, 2), cex.labels = 1, axisOpts =  
  list(xAxis = TRUE), ...)), parVals = list(mfrow = c(2, 2), xaxs = "r", yaxs  
  = "r", pty = "s", mai = c(0.2, 0.2, 0.05, 0.05)), ...)  
```


Arguments

x a vector of observations, or the residuals from fitting a linear model. Alternatively, a fitted lm object. If x is a single vector, then the implicit assumption is that the mean (or null) model is being fitted, i.e. $\text{lm}(x \sim 1)$ and that the data are best summarised by the sample mean.

plotOrder the order of the plots. 1: pred-res plot, 2: normal Q-Q plot, 3: histogram, 4: Cooks's Distance plot.

args a list containing three additional lists eovcheckArgs, normcheckArgs and cooksArgs. The elements of these lists are the optional arguments of eovcheck, normcheck and cooks20x, and are explained in more detail in those functions. Most users will never use these arguments, but they provide super-flexibility in terms of what is displayed.

parVals the values that are set via par for this plot. These are mfrow, xaxs, yaxs, pty, and mai. Most users will never use these arguments, but they provide super-flexibility in terms of what is displayed.

... additional parameters. Included for future flexibility, but unsure how this might be used currently.

Methods (by class)

- lm: Model checking plots

Examples

# An exponential growth curve
e = rnorm(100, 0, .1)
x = rnorm(100)
y = exp(5 + 3 * x + e)
fit = lm(y ~ x, data = data.frame(x, y))
modcheck(fit)

# An exponential growth curve with the correct transformation
fit = lm(log(y) ~ x, data = data.frame(x, y))
modcheck(fit)

# Peruvian Indians data
data(peru.df)
modcheck(lm(BP ~ weight, data = peru.df))

mozart.df  Length of Mozart's Movements

Description

Length of movements from 11 of Mozart’s early symphonies and 11 of his late symphonies.
**Format**

A data frame with 88 observations on 3 variables.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>.1</td>
<td>Time</td>
<td>numeric</td>
</tr>
<tr>
<td>.2</td>
<td>Movement</td>
<td>factor</td>
</tr>
<tr>
<td>.3</td>
<td>Period</td>
<td>factor</td>
</tr>
</tbody>
</table>

---

**Description**

Calculates and prints the estimate, multiple 95% confidence intervals; unadjusted, Tukey and Bonferroni p-values for all possible differences in means in a one-way ANOVA.

**Usage**

```r
multipleComp(fit, conf.level = 0.95)
```

**Arguments**

- **fit**: output from the command `lm()`.
- **conf.level**: confidence level for the confidence interval, expressed as a percentage.

**Value**

Returns a list of estimates, confidence intervals and p-values.

**Examples**

```r
## computer data
data(computer.df)
fit = lm(score ~ factor(selfassess), data = computer.df)
multipleComp(fit)
```

---

**nail.df**

**Nail Polish Data**

**Description**

These data were collected to determine whether quick drying nail polish or regular nail polish dried faster. The time for each type of nail polish to dry was recorded.

**Format**

A data frame with 60 observations on 2 variables.
normcheck

Testing for normality plot

Description

Plots two plots side by side. Firstly it draws a Normal QQ-plot of the residuals, along with a line which has an intercept at the mean of the residuals and a slope equal to the standard deviation of the residuals. If shapiro.wilk = TRUE then, in the top left hand corner of the Q-Q plot, the P-value from the Shapiro-Wilk test for normality is given. Secondly, it draws a histogram of the residuals. A normal distribution is fitted and superimposed over the histogram. NOTE: if you want to leave the x-axis blank in the histogram then, use xlab = c("Theoretical Quantiles", ""), i.e. leave a space between the quotes. If you don’t leave a space, then information will be extracted from x.

Usage

`normcheck(x, ...)`

## Default S3 method:
`normcheck(x, xlab = c("Theoretical Quantiles", ""),
  ylab = c("Sample Quantiles", ""), main = c("", ""), col = "light blue",
  bootstrap = FALSE, B = 5, bpch = 3, bcol = "lightgrey",
  shapiro.wilk = FALSE, whichPlot = 1:2, usePar = TRUE, ...)

## S3 method for class 'lm'
`normcheck(x, xlab = c("Theoretical Quantiles", ""),
  ylab = c("Sample Quantiles", ""), main = c("", ""), col = "light blue",
  bootstrap = FALSE, B = 5, bpch = 3, bcol = "lightgrey",
  shapiro.wilk = FALSE, whichPlot = 1:2, usePar = TRUE, ...)

Arguments

- `x` the residuals from fitting a linear model. Alternatively, a fitted `lm` object.
- `...` additional arguments which are passed to both `qqnorm` and `hist`
- `xlab` a title for the x-axis of both the Q-Q plot and the histogram: see `title`.
- `ylab` a title for the y-axis of both the Q-Q plot and the histogram: see `title`.
- `main` a title for both the Q-Q plot and the histogram: see `title`.
- `col` a color for the bars of the histogram.
- `bootstrap` if TRUE then B samples will be taken from a Normal distribution with the same mean and standard deviation as x. These will be plotted in a lighter colour behind the empirical quantiles so that we can see how much variation we would expect in the Q-Q plot for a sample of the same size from a truly normal distribution.
the number of bootstrap samples to take. Five should be sufficient, but hey maybe you want more?

bpch the plotting symbol used for the bootstrap samples. Legal values are the same as any legal value for pch as defined in \texttt{par}.

bcol the plotting colour used for the bootstrap samples. Legal values are the same as any legal value for col as defined in \texttt{par}.

\texttt{shapiro.test} if TRUE, then in the top left hand corner of the Q-Q plot, the P-value from the Shapiro-Wilk test for normality is displayed.

\texttt{whichPlot} legal values are 1, 2, and any pair of the two, i.e. 1:2, 2:1, c(1,2), c(2,1), or even variants of c(1,1) (although I do not know why you would) want to do this. 1:2 is used by default and returns a normal Q-Q plot and a histogram of the residuals in that order. The order of the labels in xlab and ylab assume this order, and will be reordered automatically if the order is anything other than 1:2.

\texttt{usePar} if TRUE, then this function will set \texttt{par} for the user. If FALSE, then this function assumes \texttt{par} has been set by the user and therefore should not be be over-ridden.

\textbf{Methods (by class)}

- default: Testing for normality plot
- \texttt{lm}: Testing for normality plot

\textbf{See Also}

\texttt{shapiro.test}.

\textbf{Examples}

```r
# An exponential growth curve
e = rnorm(100, 0, 0.1)
x = rnorm(100)
y = exp(5 + 3 * x + e)
fit = lm(y ~ x)
normcheck(fit)

# An exponential growth curve with the correct transformation
fit = lm(log(y) ~ x)
normcheck(fit)

# Same example as above except we use normcheck.default
normcheck(residuals(fit))

# Peruvian Indians data
data(peru.df)
normcheck(lm(BP ~ weight, data = peru.df))
```
onewayPlot  One-way Analysis of Variance Plot

Description

Displays stripplot/boxplot of the response variable with intervals by factor levels. It is used as part of a one-way ANOVA analysis.

Usage

onewayplot(x, ...)

## Default S3 method:
onewayplot(x, f, conf.level = 0.95,
    interval.type = "tukey", pooled = TRUE, strip = TRUE, vert = TRUE,
    verbose = FALSE, ylabel = deparse(terms(formula)[[2]]),
    flabel = deparse(terms(formula)[[3]]), ...)

## S3 method for class 'formula'
onewayplot(formula, data = parent.frame(),
    conf.level = 0.95, interval.type = "tukey", pooled = TRUE,
    strip = TRUE, vert = TRUE, verbose = FALSE,
    ylabel = deparse(terms(formula)[[2]]),
    flabel = deparse(terms(formula)[[3]]), ...)

## S3 method for class 'lm'
onewayplot(x, ..., ylabel = nms[1], flabel = nms[2])

Arguments

x  a vector of responses, a formula object or an lm object
...
  optional arguments.
f  if x is a vector of responses then f contains the group labels for each observation in x. That is, the ith value in f says which group the ith observation of x belongs to.
conf.level  confidence level of the intervals.
interval.type  three options for intervals appearing on plot: 'hsd','lsd' or 'ci'.
pooled  two options: pooled or unpooled standard deviation used for plotted intervals.
strip  if strip=F, boxplots are displayed instead.
vert  if vert=F, horizontal stripplots are displayed instead (boxplots can only be displayed vertically).
verbose  if true, print intervals on console.
ylabel  can be used to replace variable name of y by another string.
flabel  can be used to replace variable name of f by another string.
**Description**

Data from an experiment to determine the abundance of oysters recruiting from three sites in two different estuaries in New South Whales. One in Georges River and two in Port Stephens. The number of oysters were recorded for 10 cm by 10 cm panels over a two year period.

**Format**

A data frame with 87 observations on 2 variables.
peru.df

[1] Oysters numeric Number of oysters on each experimental panel
[2] Site factor Location of the experimental panels (GR = Georges River, PS1 = First Port Stephens Site, PS2 = Second Port Stephens Site)

---

**pairs20x**

*Pairwise Scatter Plots with Histograms and Correlations*

**Description**

Plots pairwise scatter plots with histograms and correlations for the data frame.

**Usage**

`pairs20x(x, na.rm = TRUE, ...)`

**Arguments**

- **x**: a data frame.
- **na.rm**: if TRUE then only complete cases will be displayed
- **...**: optional arguments which are passed to the generic pairs function.

**Value**

Returns the plots.

**See Also**

'pairs', 'panel.smooth', 'panel.cor', 'panel.hist'

**Examples**

```r
## Peruvian Indians
data(peru.df)
pairs20x(peru.df)
```

---

**peru.df**

*Peruvian Indians*

---

**Description**

A random sample of Peruvian Indians born in the Andes mountains, but who have since migrated to lower altitudes. The sample was collected to assess the long term effects of altitude on blood pressure.

**Format**

A data frame with 39 observations on 5 variables.
predict20x

Descriptive statistics:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>age</td>
<td>numeric  Subject’s age</td>
</tr>
<tr>
<td>[2]</td>
<td>years</td>
<td>numeric  Number of years since migration</td>
</tr>
<tr>
<td>[3]</td>
<td>weight</td>
<td>numeric  Subject’s weight (kg)</td>
</tr>
<tr>
<td>[4]</td>
<td>height</td>
<td>numeric  Subject’s height (mm)</td>
</tr>
<tr>
<td>[5]</td>
<td>BP</td>
<td>numeric  Subject’s systolic blood pressure (mm Hg)</td>
</tr>
</tbody>
</table>

Model Predictions for a Linear Model

Description

Uses the main output and some error messages from R function ‘predict’ but gives you more output. (Error messages are not reliable when used in Splus.)

Usage

`predict20x(object, newdata, cilevel = 0.95, digit = 3, print.out = TRUE, ...)`

Arguments

- `object` an `lm` object, i.e. the output from `lm`.
- `newdata` prediction data frame.
- `cilevel` confidence level of the interval.
- `digit` decimal numbers after the point.
- `print.out` if TRUE, print out the prediction matrix.
- `...` optional arguments that are passed to the generic ‘predict’

Details

Note: The data frame, newdata, must have the same column order and data types (e.g. numeric or factor) as those used in fitting the model.

Value

- `frame` vector or matrix including predicted values, confidence intervals and predicted intervals.
- `fit` prediction values.
- `se.fit` standard error of predictions.
- `residual.scale` residual standard deviations.
- `df` degrees of freedom for residual.
- `cilevel` confidence level of the interval.
Note

This function is deprecated. It will be removed in future versions of the package.
this function is deprecated as it is never used in class any more. We prefer the standard predict
method.

See Also

predict, predict.lm, as.data.frame.

Examples

# Zoo data
data(zoo.df)
zoo.df = within(zoo.df, {day.type = factor(day.type))}
zoo.fit = lm(log(attendance) ~ time + sun.yesterday + nice.day + day.type + tv.ads,
data = zoo.df)
pred.zoo = data.frame(time = 8, sun.yesterday = 10.8, nice.day = 0,
day.type = factor(3), tv.ads = 1.181)
predict20x(zoo.fit, pred.zoo)

# Peruvian Indians data
data(peru.df)
peru.fit = lm(BP ~ age + years + I(years^2) + weight + height, data = peru.df)
pred.peru = data.frame(age = 21, years = 2, `I(years^2)` = 2, weight = 71, height = 1629)
predict20x(peru.fit, pred.peru)

---

**predictCount**

Predicted Counts for a Generalized Linear Model

Description

Uses the main output and some error messages from R function 'predict' but gives you more output.
(Error messages are not reliable when used in Splus.)

Usage

predictCount(object, newdata, cilevel = 0.95, digit = 3, print.out = TRUE,
...)

Arguments

- **object**: a glm object, i.e. the output from glm.
- **newdata**: prediction data frame.
- **cilevel**: confidence level of the interval.
- **digit**: decimal numbers after the point.
- **print.out**: if TRUE, print out the prediction matrix.
- **...**: optional arguments that are passed to the generic predict.
Details

Note: The data frame, newdata, must have the same column order and data types (e.g. numeric or factor) as those used in fitting the model.

Value

A data frame with three columns:

- **Predicted** the predicted count.
- **Conf.lower** the lower bound of the predicted count.
- **Conf.upper** the upper bound of the predicted count.

See Also

- `predict`, `predict.glm`, `as.data.frame`.

---

**LSD-Display Intervals**

Description

This function is called by `rowdistr`.

Usage

```r
propslsd.new(crosstablist, conf.level = 0.95, arrowlength = 0.1)
```

Arguments

- `crosstablist`: A list produced by `crosstabs` or a matrix containing a 2-way table of counts (without marginal totals).
- `conf.level`: Confidence level of the intervals.
- `arrowlength`: Length of the arrows.

Note

This function is no longer exported as it should never be called by the user. It is also deprecated and will be removed from future versions of the package.

See Also

- `crosstabs`, `rowdistr`
Cloud Seeding and Levels of Rainfall

Description

Data from an experiment to see if seeding clouds with Silver Nitrate effects the amount of rainfall.

Format

A data frame with 50 observations on 2 variables.

```
[,1] rain  numeric  Amount of rain
[,2] seed factor  Whether the clouds are seeded or not (seeded, unseeded)
```

Source


Fitted values versus residuals plot

Description

Plots a scatter plot for the variables of the residuals and fitted values from the linear model, lmfit. A lowess smooth line for the underlying trend, as well as one standard deviation error bounds for the scatter about this trend, are added to this scatter plot. A test for a quadratic relationship between the residuals and the fitted values is also computed.

Usage

```
residPlot(lmfit, f = 0.5)
```

Arguments

- `lmfit` an `lm` object, i.e. the output from `lm`.
- `f` the smoother span. This gives the proportion of points in the plot which influence the smooth at each value. Larger values give more smoothness.

Value

Returns the plot.

Note

This function is deprecated. It will be removed in future versions of the package.
See Also
trendscatter

Examples

# Peruvian Indians data
data(peru.df)
fit=lm(BP~age+years+weight+height, data = peru.df)
residPlot(fit)

rowdistr

Row distributions from a cross-tabulation of two variables

Description

Produces summaries and plots from a cross-tabulation. The output produced depends on the parameter ‘comp’. Columns relate to response categories and rows to different populations.

Usage

rowdistr(crosstablist, comp = c("basic", "within", "between"),
         conf.level = 0.95, plot = TRUE, suppressText = FALSE)

Arguments

crosstablist a list produced by 'crosstab' or a matrix containing a 2-way table of counts
(without marginal totals).
comp three options: 'basic' (default), 'within', and 'between'.
conf.level confidence level of the intervals.
plot if FALSE then the row distribution plots are not displayed
suppressText if TRUE then text results are not displayed

Details

The 'basic’ option (default) produces the response distribution for each row population together with comparative bar charts.

If comp = 'between' the resulting output displays how the probability of falling into a response class (column) differs between populations. Confidence intervals for differences in proportions are produced together with a set of bar charts with LSD intervals.

If comp = 'within' the resulting output shows the extent to which the component probabilities of the same row distribution differ. Separate Chi-square tests for uniformity are produced for each row distribution as are confidence intervals for differences in proportions within the same distribution.

Arguments plot and suppressText are really only used when producing knitr or Sweave documents so that just the plot or just the text can be displayed in the document.
Value

A matrix of row proportions, i.e., cell counts divided by row marginals.

See Also

crosstabs

Examples

data(body.df)
z = crosstabs(~ ethnicity + married, data = body.df)
rowdistr(z)
rowdistr(z, comp='between')
rowdistr(z, comp='within')

# from matrix of counts
z = matrix(c(4,3,2,6,47,20,40,62,11,8,7,22,3,0,1,10), 4, 4)
rowdistr(z)

rr

Read Data

Description

For internal use

Usage

rr()

s0x

s20x: Functions for University of Auckland Course STATS 201/208 Data Analysis

Description

A set of functions used in teaching STATS 201/208 Data Analysis at the University of Auckland. The functions are designed to make parts of R more accessible to a large undergraduate population who are mostly not statistics majors.

Author(s)

Andrew Balemi, Dineika Chandra, James Curran, Brant Deppa, Mike Forster, Brendan McArdle, Chris Wild
**seeds.df**

*Seeds Data*

**Description**

These data record the number of seeds (out of 100) that germinated when given different amounts of water. The seeds were either exposed to light or kept in the dark. Four identical boxes were used for each combination of water and light.

**Format**

A data frame with 48 observations on 3 variables.

- [.1] **Light** integer Seeds exposed to light (N=No, Y=Yes)
- [.2] **Water** integer Amount of water, higher levels correspond to more water (1, 2, 3, 4, 5, 6)
- [.3] **Count** integer Number of seeds that germinated (out of 100)

**sheep.df**

*Sheep Data*

**Description**

Sheep Data

**Format**

A data frame with 100 observations on 3 variables.

- [.1] **Weight** integer
- [.2] **Copper** factor levels (No, Yes)
- [.3] **Cobalt** factor levels (No, Yes)

**skewness**

*Skewness Statistic*

**Description**

Calculates the skewness statistic of the data in 'x'. Values close to zero correspond to reasonably symmetric data, positive values of this measure indicate right-skewed data whereas negative values indicate left-skewness.

**Usage**

skewness(x)
skulls.df

Arguments

x  vector containing the data.

Value

Returns the value of the skewness.

Examples

```r
## Mergers data:
data(mergers.df)
skewness(mergers.df$mergerdays)
```

---

<table>
<thead>
<tr>
<th>skulls.df</th>
<th><strong>Skulls Data</strong></th>
</tr>
</thead>
</table>

Description

Male Egyptian skulls from five different epochs. Each skull has had four measurements taken of it, BH, Basibregmatic Height, BL, Basialveolar Length, MB, Maximum Breadth and NH, Nasal Height. It is of interest to investigate the change in shape over time. A gradual change, would indicate inbreeding of the populations. This data only includes the maximum breadth measurements.

Format

A data frame with 150 observations on 2 variables.

```
[,1] measurement  integer
[,2] year           integer
```

Source

A Handbook of Small Data Sets

References


soyabean.df  Soya Bean Yields

Description

Data from an experiment to examine the effects of different planting times on the yield of soya beans, given four different cultivars.

Format

A data frame with 32 observations on 3 variables.

<table>
<thead>
<tr>
<th></th>
<th>yield numeric</th>
<th>cultivar factor</th>
<th>planttime factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yield of each plant</td>
<td>Cultivar used (cult1, cult2, cult3, cult4)</td>
<td>Month of planting (Novemb, Decemb)</td>
</tr>
</tbody>
</table>

Source

Littler, R. University of Waikato

stripqq  Strip charts and normal quantile-quantile plots

Description

Draws strip charts and normal quantile quantile plots of x for each value of the grouping variable g

Usage

stripqq(formula, ...)

## S3 method for class 'formula'
stripqq(formula, data = NULL, ...)

Arguments

- formula: A symbolic specification of the form x ~ g can be given, indicating the observations in the vector ‘x’ are to be grouped according to the levels of the factor ‘g’. NAs are allowed in the data.
- data: An optional data frame in which to evaluate the formula
- ...: Optional arguments that are passed to the stripchart function.

Methods (by class)

- formula: Strip charts and normal quantile-quantile plots
**Note**

This function is deprecated and will be removed in later versions of the package.

**Examples**

```r
## Zoo data
data(zoo.df)
stripqattendance~day.type, data = zoo.df)
```

---

**summary1way**  
*One-way Analysis of Variance Summary*

**Description**

Displays summary information for a one-way anova analysis. The lm object must come from a numerical response variable and a single factor. The output includes: (i) anova table; (ii) numeric summary; (iii) table of effects; (iv) plot of data with intervals.

**Usage**

```r
summary1way(fit, digit = 5, conf.level = 0.95, inttype = "tukey",
pooled = TRUE, print.out = TRUE, draw.plot = TRUE, ...)
```

**Arguments**

- `fit` an lm object, i.e. the output from `lm`.
- `digit` decimal numbers after the point.
- `conf.level` confidence level of the intervals.
- `inttype` three options for intervals appeared on plot: 'hsd', 'lsd' or 'ci'.
- `pooled` two options: pooled or unpooled standard deviation used for plotted intervals.
- `print.out` if TRUE, print out the output on the screen.
- `draw.plot` if TRUE, plot data with intervals.
- `...` more options.

**Value**

- `df` degrees of freedom for regression, residual and total.
- `Sum of Sq` sum squares for regression, residual and total.
- `Mean Sq` mean squares for regression and residual.
- `F value` F-statistic value.
- `Pr(F)`
- `Main Effect` Group Effects
See Also

summary2way, anova, aov, dummy.coef, onewayPlot

Examples

```r
attitudes = c(5.2,5.2,6.1,6.5,6.75,5.6,6.25,6.8,6.87,7.1,
              6.3,6.35,5.5,5.75,4.6,5.36,5.85,5.9)
l = rep(c('Gp1','Gp2','Gp3'),rep(6,3))
l = factor(l)
f = lm(attitudes ~ l)
result = summary2way(f)
result
```

**summary2way**  

Two-way Analysis of Variance Summary

Description

Displays summary information for a two-way anova analysis. The lm object must come from a numerical response variable and factors. The output depends on the value of page:

Usage

```r
summary2way(fit, page = c("table", "means", "effects", "interaction",
                           "nointeraction"), digit = 5, conf.level = 0.95, print.out = TRUE,
             new = TRUE, all = FALSE, ...)
```

Arguments

- `fit`: an lm object, i.e. the output from `lm()`.
- `page`: options for output: 'table', 'means', 'effects', 'interaction', 'nointeraction'
- `digit`: the number of decimal places in the display.
- `conf.level`: confidence level of the intervals.
- `print.out`: if TRUE, print out the output on the screen.
- `new`: if TRUE then this will run the new version of summary2way which should be more robust than the old version. It does not work in the same way however. In particular, when page = 'means' it does not return summary statistics for each grouping of the data (pooled/by row factor/by column factor/by interaction factor). Instead it simply returns the means for each grouping.
- `all`: Only applicable to page = "interaction". If TRUE, pairwise comparisons for all combinations of factor levels are shown. Otherwise, comparisons are only shown between combinations that have the same level for one of the factors.
- `...`: other arguments like inttype, pooled etc.
Details

page = 'table' anova table page = 'means' cell means matrix, numeric summary page = 'effects'
table of effects page = 'interaction' tables of contrasts page = 'nointeraction' tables of contrasts

Value

A list with the following components:

- **Df**: degrees of freedom for regression, residual and total.
- **Sum of Sq**: sum squares for regression, residual and total.
- **Mean Sq**: mean squares for regression and residual.
- **F value**: F-statistic value.
- **Pr(F)**: The P-value associated with each F-test.
- **Grand Mean**: The overall mean of the response variable.
- **Row Effects**: The main effects for the first (row) factor.
- **Col Effects**: The main effects for the second (column) factor.
- **Interaction Effects**: The interaction effects if an interaction model has been fitted, otherwise NULL.

- **results**: If new = TRUE, then this is a list with five components: table - the ANOVA
table, means - the table of means from model.tables, effects - the table of
effects from model.tables, and comparisons - the differences in the means
with standard errors, confidence bounds, and P-values from TukeyHSD

See Also

summary2way, model.tables, TukeyHSD

Examples

```r
# Arousal data:
data(arousal.df)
arousal.fit = lm(arousal ~ gender * picture, data = arousal.df)
summary2way(arousal.fit)
```
**Description**

Produces a table of summary statistics for the data. If the argument group is missing, calculates a matrix of summary statistics for the data in \( x \). If group is present, the elements of group are interpreted as group labels and the summary statistics are displayed for each group separately.

**Usage**

```r
summaryStats(x, ...)  
## Default S3 method:  
summaryStats(x, group = rep("Data", length(x)),  
data.order = TRUE, digits = 2, ...)  
## S3 method for class 'formula'  
summaryStats(x, data = NULL, data.order = TRUE,  
digits = 2, ...)  
## S3 method for class 'matrix'  
summaryStats(x, data.order = TRUE, digits = 2, ...)```

**Arguments**

- **x** either a single vector of values, or a formula of the form data~group, or a matrix.
- **...** Optional arguments.
- **group** a vector of group labels.
- **data.order** if TRUE, the group order is the order which the groups are first encountered in the vector 'group'. If FALSE, the order is alphabetical.
- **digits** the number of decimal places to display.
- **data** an optional data frame containing the variables in the model.

**Value**

If \( x \) is a single variable, i.e. there are no groups, then a single list is invisibly returned with the following named items:

- **min** Minimum value.
- **max** Maximum value.
- **mean** Mean value.
- **var** Variance – the average of the squares of the deviations of the data values from the sample mean.
- **sd** Standard deviation – the square root of the variance.
summaryStats

- **n**  Number of data values – size of the data set.
- **iqr**  Midspread (IQR) – the range spanned by central half of data; the interquartile range.
- **skewness**  Skewness statistic – indicates how skewed the data set is. Positive values indicate right-skew data. Negative values indicate left-skew data.
- **lq**  Lower quartile
- **median**  Median – the middle value when the batch is ordered.
- **uq**  Upper quartile

If grouping is provided, either by using the group argument, or providing a factor in a formula, or by passing a matrix where the different columns represent the groups, then the function will return a data.frame a row containing all the statistics above for each group.

**Methods (by class)**

- default: Summary Statistics
- formula: Summary Statistics
- matrix: Summary Statistics

**Examples**

```r
## STATS20x data:
data(course.df)

## Single variable summary
with(course.df, summaryStats(Exam))

## Using a formula
summaryStats(Exam ~ Stage1, course.df)

## Using a matrix
X = cbind(rnorm(50), rnorm(50))
summaryStats(X)

## Saving and extracting the information
sumStats = summaryStats(Exam ~ Degree, course.df)
sumStats

## Just the BAs
sumStats['BA', ]

## Just the means
sumStats$mean
```
teach.df  

Comparison of Three Teaching Methods

Description

Data from an experiment to assess the impact of three different teaching methods on language ability. 30 students were randomly allocated into three groups, one for each method. The students’ IQ before instruction and a language test score after instruction were recorded.

Format

A data frame with 30 observations on 3 variables.

[.1] lang  numeric  Language test score after instruction
[.2] IQ    numeric  Student’s IQ
[.3] method factor  Teaching method (1, 2, 3)

technitron.df  

Technitron Salary Information

Description

Salary information for all salaried employees of the Technitron Company.

Format

A data frame with 46 observations on 8 variables.

[.1] salary  numeric  Annual Salary ($)
[.2] yrs.empl numeric  Number of years employed at Technitron
[.3] prior.yrs numeric  Number of years prior experience
[.4] edu     numeric  Years of education after high school
[.5] id      numeric  Company identification number
[.6] gender  numeric  Gender (0 = female, 1 = male)
[.7] dept    numeric  Department employee works in (1 = Sales, 2 = Purchasing, 3 = Advertising, 4 = Engineering)
[.8] super   numeric  Number of employees supervised

thyroid.df  

Effect of a New Drug on Thyroid Weights
Description

Data from an experiment to assess the effect of a new drug on the weight of the thyroid gland using 16 laboratory animals. The animals were randomly assigned into either a control group, or a treatment group, and each animal had its bodyweight recorded at the beginning of the experiment and its thyroid weight measured at the end of the experiment.

Format

A data frame with 16 observations on 3 variables.

- [.] thyroid numeric Weight of thyroid gland after 7 days (mg)
- [.] body numeric Animal body weight before experiment began (g)
- [.] group factor Animal’s group (1 = control, 2 = drug)

---
toothpaste.df Crest Toothpaste

Description

Two random samples of households, one of households who purchase Crest toothpaste and one of households who do not. For each household the age is recorded of the person responsible for purchasing the toothpaste.

Format

A data frame with 20 observations on 2 variables.

- [.] purchasers numeric Age of the person in the household responsible for purchases of Crest
- [.] nonpurchasers numeric Age of the person in the household responsible for purchases of other brands of toothpaste

---
trendscatter Trend and scatter plot

Description

Plots a scatter plot for the variables x, y along with a lowess smooth for the underlying trend. One standard deviation error bounds for the scatter about this trend are also plotted.

Usage

trendscatter(x, ...)

## Default S3 method:
trendscatter(x, y = NULL, f = 0.5, xlab = NULL, ylab = NULL, main = NULL, ...)

## S3 method for class 'formula'
trendscatter(x, f = 0.5, data = NULL, xlab = NULL, ylab = NULL, main = NULL, ...)

### Arguments

- `x`: the coordinates of the points in the scatter plot. Alternatively, a formula.
- `...`: Optional arguments
- `y`: the y coordinates of the points in the plot, ignored if `x` is a function.
- `f`: the smoother span. This gives the proportion of points in the plot which influence the smooth at each value. Larger values give more smoothness.
- `xlab`: a title for the x axis: see `title`.
- `ylab`: a title for the y axis: see `title`.
- `main`: a title for the plot: see `title`.
- `data`: an optional data frame containing the variables in the model.

### Value

Returns the plot.

### Methods (by class)

- default: Trend and scatter plot
- formula: Trend and scatter plot

### See Also

- `residPlot`

### Examples

```r
# A simple polynomial
x = rnorm(100)
e = rnorm(100)
y = 2 + 3 * x - 2 * x^2 + 4 * x^3 + e
trendscatter(y ~ x)

# An exponential growth curve
e = rnorm(100, 0, 0.1)
y = exp(5 + 3 * x + e)
trendscatter(log(y) ~ x)

# Peruvian Indians data
data(peru.df)
```
trendscatter(BP ~ weight, data = peru.df)

# Note: this usage is deprecated
with(peru.df,trendscatter(weight,BP))

---

zoo.df  
Zoo Attendance during an Advertising Campaign

---

Description

Data for 455 days of attendance records for Auckland Zoo, from January 1, 1993. Note that only 440 values are given due to missing values. It was of interest to assess whether an advertising campaign was effective in increasing attendance.

Format

A data frame with 440 observations on 6 variables.

<table>
<thead>
<tr>
<th></th>
<th>attendance</th>
<th>numeric</th>
<th>Number of visitors</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>time</td>
<td>numeric</td>
<td>Time in days since the start of the study</td>
</tr>
<tr>
<td>[3]</td>
<td>sun.yesterday</td>
<td>numeric</td>
<td>Hours of sunshine the previous day</td>
</tr>
<tr>
<td>[4]</td>
<td>tv.adds</td>
<td>numeric</td>
<td>Average spending on TV advertising in the previous week ($000 per day)</td>
</tr>
<tr>
<td>[5]</td>
<td>nice.day</td>
<td>factor</td>
<td>Assessment based on number of hours of sunshine (0 = No, 1 = Yes)</td>
</tr>
<tr>
<td>[6]</td>
<td>day.type</td>
<td>factor</td>
<td>Type of day (1 = ordinary weekday, 2 = weekend day, 3 = school holiday weekday, 4 = public holiday)</td>
</tr>
</tbody>
</table>
Index

*Topic datasets
  airpass.df, 3
  apples.df, 3
  arousal.df, 4
  beer.df, 5
  body.df, 5
  books.df, 5
  bursary.df, 7
  butterfat.df, 7
  camplake.df, 8
  chalk.df, 8
  computer.df, 9
  course.df, 11
  course2way.df, 11
  diamonds.df, 14
  fire.df, 17
  fruitfly.df, 19
  house.df, 20
  incomes.df, 20
  lakemary.df, 22
  larain.df, 22
  mazda.df, 24
  mening.df, 25
  mergers.df, 25
  mozart.df, 26
  nail.df, 27
  oysters.df, 31
  peru.df, 32
  rain.df, 36
  seeds.df, 39
  sheep.df, 39
  skulls.df, 40
  soyabean.df, 41
  teach.df, 47
  technitron.df, 47
  thyroid.df, 47
  toothpaste.df, 48
  zoo.df, 50

*Topic debugging
  getVersion, 19

*Topic device
  layout20x, 23

*Topic hplot
  autocorPlot, 4
  boxqq, 6
  cooks20x, 10
  eovcheck, 14
  interactionPlots, 20
  modcheck, 25
  normcheck, 28
  onewayPlot, 30
  pairs20x, 32
  residPlot, 36
  stripqq, 41
  trendscatter, 48

*Topic htest
  cireg, 9
  crosstabs, 13
  freq1way, 17
  levene.test, 23
  multipleComp, 27
  predict20x, 33
  predictCount, 34
  propslsd.new, 35
  rowdistr, 37

*Topic models
  crossFactors, 12
  estimateContrasts, 16
  rr, 38
  summary1way, 42
  summary2way, 43

*Topic multivariate
  summaryStats, 45

*Topic univar
  skewness, 39

airpass.df, 3
anova, 9, 24, 43
aov, 43
apples.df, 3
arousal.df, 4
as.data.frame, 34, 35
autocor.plot (autocorPlot), 4
autocorPlot, 4
beer.df, 5
body.df, 5
books.df, 5
boxqq, 6
bursary.df, 7
butterfat.df, 7
camplake.df, 8
chalk.df, 8
ciReg, 9
computer.df, 9
cooks2x, 10, 26
course.df, 11
course2way.df, 11
crossFactors, 12, 24
crosstabs, 13, 35, 38
diamonds.df, 14
dummy.coef, 43
eovcheck, 14, 26
estimateContrasts, 16
factor, 12
fire.df, 17
dfreq1way, 17
fruitfly.df, 19
getVersion, 19
glm, 34
house.df, 20
incomes.df, 20
interactionPlots, 20
lakemary.df, 22
larain.df, 22
layout2x, 23
levene.test, 15, 23
lm, 9, 10, 42
mazda.df, 24
mening.df, 24
mergers.df, 25
modcheck, 25
model.tables, 44
mozart.df, 26
multipleComp, 17, 27
nail.df, 27
normcheck, 26, 28
onewayPlot, 30, 43
oysters.df, 31
pairs2x, 32
par, 6, 26, 29
peru.df, 32
predict, 34, 35
predict.glm, 35
predict.lm, 34
predict2x, 33
predictCount, 34
propsld.new, 35
rain.df, 36
residPlot, 36, 49
rowdistr, 35, 37
rr, 38
s0x, 38
s0x-package (s0x), 38
seeds.df, 39
shapiro.test, 29
sheep.df, 39
skewness, 39
skulls.df, 40
soyabean.df, 41
stripqq, 41
summary, 9
summary1way, 17, 31, 42, 44
summary2way, 17, 22, 43, 43
summaryStats, 45
t.test, 31
teach.df, 47
technitron.df, 47
thyroid.df, 47
title, 15, 28, 49
toothpaste.df, 48
trendscatter, 37, 48
TukeyHSD, 44
twosampPlot (onewayPlot), 30
INDEX

zoo.df, 50