Package 'handyplots'

July 22, 2025

Type Package

Title Handy Plots

Version 1.1.3

Date 2019-01-11

Author Jonathan Schwartz

Maintainer Jonathan Schwartz <jzs1986@gmail.com>

Description Several handy plots for quickly looking at the relationship between two numeric vectors of equal length. Quickly visualize scatter plots, residual plots, qq-plots, box plots, confidence intervals, and prediction intervals.

License GPL (>= 2)

Imports stats, graphics

Depends R (>= 3.4)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-19 23:40:10 UTC

Contents

---

ciplot   *Confidence Interval Plot*

Description

given two numeric vectors of equal length, plot a scatter plot of the data, the regression line, and
a confidence interval for the mean of a new observation or the prediction interval for a single new
observation.

Usage

```
ciplot(x, y, x0 = NULL, int = c("p","c"), level = 0.95,
relationship = c("linear","quadratic","cubic","sqrt","exponential","reciprocal","log"),
show.range = TRUE, user.xlim = NULL, user.ylim = NULL)
```

Arguments

x	a numeric vector of length > 3
y	a numeric vector of length > 3 (equal in length to x)
x0	the x value at which you wish to make a prediction (NULL by default. If left blank, the plot will not show a prediction at a particular x value.)
int	interval type. "prediction" by default (can be abbreviated), which will plot the prediction interval for a single new observation. If specified with "confidence" (can be abbreviated), it will show the confidence interval for the mean value of a new observation.
level	the confidence level at which you wish to predict. 0.95 by default. If you wish to specify a confidence level, it must be a numerical value greater than 0 and less than 1.
relationship	the type of relationship that the two vectors share. "linear" by default. May specify a different type of relationship with "quadratic" (may be abbreviated to "quad"), "cubic", "sqrt", "exponential" (may be abbreviated to "exp"), "reciprocal" (may be abbreviated to "recip"), or "logarithmic" (may be abbreviated to "log"). Specifying a different type of relationship will change the shape of the regression line.
show.range	logical. If TRUE (the default), dotted red lines will show the confidence or prediction interval along the entire plot. If FALSE, it will only show the confidence/prediction interval at a specified x value (if x0 is set).
user.xlim	the interval of x values the user wishes to display in the plot. If left unspecified, it will be NULL and the default x limits will be plotted (which will be the entire range of x, including x0).
user.ylim	the interval of y values the user wishes to display in the plot. If left unspecified, it will be NULL and the default y limits will be plotted (which will be the entire range of y, including the predicted y value at x0).

Warning

If x0 is outside the domain of x, ciplot will extrapolate the data and predict a value of yhat for the given x0. This may be dangerous, depending on how your data behaves outside the existing domain.

Author(s)

Jonathan Schwartz

References

Montgomery, D. C., Peck, E. A., Vining, G. G. (2013), Introduction to Linear Regression Analysis, Hoboken, NJ: John Wiley & Sons, Inc.

See Also

plot, lm, predict

Examples

```
##predicting the mean petal width of an iris whose petal length is 2.5
ciplot(iris$Petal.Length,iris$Petal.Width,x0=2.5,int="conf")

##predicting a single new observation of the petal width of an iris whose petal length is 2.5
ciplot(iris$Petal.Length,iris$Petal.Width,x0=2.5,int="pred")

##extrapolating the data to predict the mean of the width of an iris's petal whose petal length is 8
ciplot(iris$Petal.Length,iris$Petal.Width,x0=8,int="conf")

##zooming in to the previous graph and removing the dotted red lines
ciplot(iris$Petal.Length,iris$Petal.Width,x0=8,int="conf",show.range=FALSE,
user.xlim=c(7.5,8.5),user.ylim=c(2.6,3.2))
```

---

colID   *Column ID*

Description

A quick way to see the name and class of every colum of a data frame

Usage

```
colID(df)
```

Arguments

df   A data frame you wish to look at

Value

Returns a data frame where column 1 is the names of the columns of the original data frame, and column 2 is the class of the column of the original data frame.

Author(s)

Jonathan Schwartz

See Also

data.frame, class, colnames

Examples

```
colID(iris)
```

---

fakedata   *Fake Data*

Description

A quick way to cook up some fake data.

Usage

```
fakedata(formula, s = 0.25)
```

Arguments

formula   A formula which describes the relationship you wish your fake data to have to an existing numeric vector. For example, if you have a numeric vector x, if you want your fake data to have a perfect 1-to-1 linear relationship with x, the formula would simply be x.

s   A numeric value which describes the amount of variablity you want your fake data to have. If s = 0, then the data will have no variablity at all. (i.e. the residuals will have mean 0 and variance 0). If s > 1, the data will look very scattered and random, the correlation between your existing vector and your fake data will be low.

Details

Quickly cooking up fake data may be useful for experimenting with differnt plotting functions in R with data that you can control. You can control the relationship between your data and an existing vector, and you can control the variablity of the data, i.e. how closely correlated the fake data is to the existing vector. You also know that the residuals are normally distributed with mean 0, which satisfies a major assumption of linear regression.

Value

The function returns a numeric vector.

Author(s)

Jonathan Schwartz

See Also

[rnorm](), [plot](), [lm]()

Examples

```
x=sample(0:1000,100)
y=fakedata(3*x+10) #y is a vector of fake data which will have a linear relationship with x
plot(x,y)
cor(x,y) #x and y are very highly correlated
y2=fakedata(3*x+10,1) #increasing the value of s decreases the correlation
plot(x,y2)
cor(x,y2) #x and y2 are not as highly correlated

##you can also, of course do non-linear relationships
y3=fakedata(sqrt(1/x))
plot(x,y3)
```

---

quickplot   *Quick Plot*

Description

If you have two numeric vectors of equal length you can use quickplot to quickly look at the potential relationship between them in four graphs at once.

Quickplot will show you a scatter plot with a regression line, a qq-plot to check the normality of the residuals, a residual plot to check the constancy and correlation of the residuals, and a boxplot for a quick overview of the spread of the two vectors, and two historgrams to see the distributions of the two vectors.

Usage

```
quickplot(x, y)
```

Arguments

x	A numeric vector of length > 3
y	A numeric vector of length > 3 (equal in length to x)

Author(s)

Jonathan Schwartz

References

Montgomery, D. C., Peck, E. A., Vining, G. G. (2013), Introduction to Linear Regression Analysis, Hoboken, NJ: John Wiley & Sons, Inc.

See Also

[plot](), [abline](), [lm](), [qqnorm](), [qqline](), [resplot](), [boxplot]()

Examples

```
##quickly looking at the relationship between iris petal length and iris petal width
quickplot(iris$Petal.Length,iris$Petal.Width)
```

---

resplot	*Residual Plot*

Description

Plot the fitted values vs the studentized or standardized residuals for a `glm` or `lm` object.

Usage

```
resplot(model, zoom = NULL, highlight.outliers = FALSE,
  residuals = c("student","standard"))
```

Arguments

model	a regression model with any number of predictors. Must be a `glm` or `lm` object.
zoom	what range of residuals you wish to show in your plot. By default, zoom is `NULL`, and the residual plot will show all residuals. If you set zoom to a numeric value > 0, resplot will only show residuals which are at most that many standard deviations away from 0.
highlight.outliers	
	logical. If `FALSE` (the default), outliers will not be highlighted. If `TRUE`, every residual which is more than 3 standard deviations from 0 will be circled in red.
residuals	which type of residuals to use. Studentized residuals are used by default, but can be specified with `"student"`, `"rstudent"`, or `"studentized"`. Standardized residuals can be specified with `"standard"`, `"rstandard"`, or `"standardized"`

Details

A residual plot shows the fitted values of the response variable on the x-axis and the studentized or standardized residuals on the y-axis. It can be used to check for correlated residuals or non-constant variance of the residuals, both of which would violate the residual assumptions of a linear model. It can also be used to check for outliers, as a value below -3 or above 3 would indicate a residual which is more than 3 standard deviations from the mean of 0.

Author(s)

Jonathan Schwartz

References

Montgomery, D. C., Peck, E. A., Vining, G. G. (2013), Introduction to Linear Regression Analysis, Hoboken, NJ: John Wiley & Sons, Inc.

See Also

plot, abline, lm, glm, predict, rstudent, rstandard

Examples

```
##plot a residual plot to check the model assumptions for a linear
##model of iris petal length as a predicted by iris petal width
model<-lm(iris$Petal.Length~iris$Petal.Width)
resplot(model)

##highlight the one outlier
resplot(model,highlight.outliers=TRUE)

##zoom in to only show the residuals between -1 and 1
resplot(model,zoom=1)
```

---

wordcount	*Word Count*

Description

The function takes a text file or text string and outputs a barplot of the most frequently occuring words.

Usage

```
wordcount(file = "", n, decreasing = TRUE, text)
```

Arguments

file	A text file whose location is interpreted relative to the current working directory (given by getwd). Can be left blank, in which case the text must be inputted as the argument text.
n	The number of words to show in the barplot. Should be an integer greater than 0 and less than the number of unique words in the text. For example, if n=5, then the barplot will show the 5 most frequently occuring words.
decreasing	If TRUE (the default), the words in the barplot will show from most frequent on the left to least frequent on the right. If FALSE, the most frequently occuring word will be on the right hand side of the barplot.
text	If you wish to enter text as an inline argument rather than as a file on your computer, you can enter your text as this argument and leave file blank.

Author(s)

Jonathan Schwartz

See Also

scan, barplot

Examples

```
myfile <- file.path(tempdir(), "wordcounttest.txt")
write("Four four four four. Three three three. Two two. One.",file=myfile )
wordcount(myfile ,4)

##or text can be entered inline
wordcount(text="Four four four four. Three three three. Two two. One.",n=4)
```

Index