

# Package ‘plot.counts’

March 10, 2021

**Type** Package

**Title**

Utility functions for elementary analysis and plots of cell counts under different conditions

**Version** 1.0

**Date** 2012-07-22

**Author** Thomas Braschler

**Maintainer** Thomas Braschler <thomas.braschler@gmail.com>

**Description**

This package contains convenience implementations for plotting, particularly for count data.

**License** GPL-3 — file LICENCE

**LazyLoad** yes

## R topics documented:

plot.counts-package . . . . .	2
barplot_dots . . . . .	2
barplot_with_errorbars . . . . .	3
barplot_xpos . . . . .	4
change_diagram . . . . .	5
errorbars . . . . .	6
get_associated_values . . . . .	6
get_t_test_matrix . . . . .	8
multipleHistograms . . . . .	9
pie_with_errorbars . . . . .	10
plot_counts . . . . .	11
plot_dot_column . . . . .	12
sd_mean . . . . .	13
significance_labels . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

plot.counts-package	<i>plot.counts</i>
---------------------	--------------------

---

## Description

Utility functions for comparing and elementary plotting of count data under different conditions

## Details

Package:	plot.counts
Type:	Package
Version:	1.0
Date:	2012-07-22
License:	What license is it under?
LazyLoad:	yes

## Author(s)

Thomas Braschler

Maintainer: thomas.braschler@gmail.com

---

barplot_dots	<i>barplot_dots</i>
--------------	---------------------

---

## Description

Bar plots for count data with errorbars and significance levels

## Usage

```
barplot_dots(treatment,data,main=NULL,xlab=NULL,ylab=NULL,sig_codes=NULL,ylim=NULL,xpd=NA,cex.ax
```

## Arguments

<b>treatment</b>	Vector describing the conditions applied. The distinct values in this vector will be the labels of the bars.
<b>data</b>	Vector describing the individual results corresponding to the <b>treatment</b> conditions. Both <b>treatment</b> and <b>data</b> should be the raw data, not aggregated.
<b>main</b>	Title of graphics (graphical parameter as described in <a href="#">par</a> )
<b>xlab</b>	Label of the x-axis (graphical parameter as described in <a href="#">par</a> )
<b>ylab</b>	Label of the y-axis (graphical parameter as described in <a href="#">par</a> )

<code>sig.codes</code>	Significance labels to be shown above the bars, as many entries as there are unique values in <code>treatment</code>
<code>ylim</code>	Limits of the y-axis (graphical parameter as described in <a href="#">par</a> )
<code>xpd</code>	Clipping (graphical parameter as described in <a href="#">par</a> ). Namely, supply <code>xpd=FALSE</code> for clipping of bars if <code>ylim</code> is provided with a range not starting at zero.
<code>cex.axis</code>	Character size for the axis (graphical parameter as described in <a href="#">par</a> )
<code>cex.lab</code>	Character size for the labels (graphical parameter as described in <a href="#">par</a> )
<code>las</code>	Label orientation for axis (graphical parameter as described in <a href="#">par</a> )
<code>sig.cex</code>	Character expansion for the significance label (see <a href="#">barplot_with_errorbars</a> , used internally)
<code>cex</code>	General character expansion (see <a href="#">par</a> )
<code>...</code>	Additional graphical parameters, to be passed specifically to <a href="#">plot.dot.column</a> , used internally.

**Author(s)**

Thomas Braschler

---

`barplot_with_errorbars` *barplot\_with\_errorbars*

---

**Description**

Bar plots for count data with errorbars and significance levels

**Usage**

```
barplot_with_errorbars(height, sd_height=NULL, beside = FALSE, horiz = FALSE, group_order=NULL, sig_
```

**Arguments**

<code>height</code>	As in <code>barplot</code> : Either a vector or matrix of values describing the bars which make up the plot. If <code>height</code> is a vector, the plot consists of a sequence of rectangular bars with heights given by the values in the vector. If <code>height</code> is a matrix and <code>beside</code> is <code>FALSE</code> then each bar of the plot corresponds to a column of <code>height</code> , with the values in the column giving the heights of stacked sub-bars making up the bar. If <code>height</code> is a matrix and <code>beside</code> is <code>TRUE</code> , then the values in each column are juxtaposed rather than stacked.
<code>sd_height</code>	Errors associated with the values, should have the same dimensions as <code>height</code>
<code>beside</code>	As in <code>barplot</code> : a logical value. If <code>FALSE</code> , the columns of <code>height</code> are portrayed as stacked bars, and if <code>TRUE</code> the columns are portrayed as juxtaposed bars.
<code>horiz</code>	As in <code>barplot</code> : a logical value. If <code>FALSE</code> , the bars are drawn vertically with the first bar to the left. If <code>TRUE</code> , the bars are drawn horizontally with the first at the bottom.

group_order	If provided, allows to reorder the columns of <b>height</b> and hence to change the order of the bars
sig_codes	Significance codes (or any other text) to be shown above the bars
sig.cex	Character expansion (cex) for the significance labels. Relevant only if <b>sig_codes</b> is supplied
...	Additional graphical parameters, such as <b>xlab</b> , <b>ylab</b> , will be passed to <b>barplot</b> , or, like <b>code</b> or <b>angle</b> , to the <a href="#">arrows</a> function used to produce the errorbars

### Author(s)

Thomas Braschler

---

barplot_xpos	<i>barplot_xpos</i>
--------------	---------------------

---

### Description

Draws a barplot with the bars at specific x-locations rather than regularly spread

### Usage

```
barplot_xpos(height, xpos=NULL,width = 0.5,col = NULL, border = par("fg"), main=NULL, sub=NULL, xlab)
```

### Arguments

height	The heights of the bars, should be a vector
xpos	The central positions of the bars, should be a vector of length identical to the <b>heights</b> vector
width	Width of the bars, can be either a single number or a vector of length identical to the <b>heights</b> vector
col	Color of the bars, either single character string or vector of length identical to the <b>heights</b> vector
border	Border color of the bars
main	Main title of the plot
sub	Subtitle of the plot
xlab	Label for the x-axis
ylab	Lable for the y-axis
xlim	Limits for the x-axis, numeric vector of length 2
ylim	Limits for the y-axis, numeric vector of length 2
axes	Logical indicating whether axes should be drawn
cex.axis	expansion factor for numeric axis labels (whatever that's supposed to mean, see <a href="#">barplot</a> )
plot.new	Whether the bars should be added on an existing plot or whether a new plot should be drawn

<code>density</code>	If provided, density of the hashing lines
<code>angle</code>	If density is provided, angle of the hashing lines
<code>at_x</code>	If provided, tick locations for the x-axis (see <a href="#">axis</a> )
<code>at_y</code>	If provided, tick locations for the y-axis (see <a href="#">axis</a> )
<code>labels_x</code>	If provided, tick labels for the x-axis (see <a href="#">axis</a> )
<code>labels_y</code>	If provided, tick locations for the y-axis (see <a href="#">axis</a> )
<code>...</code>	Additional graphical parameters passed on to <a href="#">plot.window</a> , <a href="#">title</a> or <a href="#">axis</a>

**Author(s)**

Thomas Braschler

---

<code>change_diagram</code>	<i>change_diagram</i>
-----------------------------	-----------------------

---

**Description**

Diagram indicating the change from initial to final state by dots connected by lines

**Usage**

```
change_diagram(y_initial,y_final,x_initial=NULL,x_final=NULL,groups=NULL,plot.new=TRUE,...)
```

**Arguments**

<code>y_initial</code>	Initially measured values, should be a numeric vector.
<code>y_final</code>	Final measured values, should be numeric vector of length identical to <code>y_initial</code> .
<code>x_initial</code>	Initial x-values. Not mandatory, but if provided, should be numeric vector of the same length as <code>y_initial</code> .
<code>x_final</code>	Final x-values. Not mandatory, but if provided, should be numeric vector of the same length as <code>y_initial</code> .
<code>groups</code>	Group identifier. Not mandatory, but if provided, should be numeric vector, character vector or factor of the same length as <code>y_initial</code> .
<code>plot.new</code>	Start a new plot (as opposed to drawing onto some pre-existing plot
<code>...</code>	Additional graphical parameters, to be passed specifically to <a href="#">plot</a> and <a href="#">lines</a> used internally. If <code>pch</code> , <code>bg</code> or <code>col</code> (see <a href="#">par</a> ) are provided, they can be vectors. If in addition a <code>groups</code> vector is provided, the values are used within the specific groups.

**Author(s)**

Thomas Braschler

---

<code>errorbars</code>	<i>errorbars</i>
------------------------	------------------

---

**Description**

Add errorbars to a plot

**Usage**

```
errorbars(x, y, sd_y = 0, angle = 90, code = 3, horiz = FALSE, ...)
```

**Arguments**

<code>x</code>	The x-values
<code>y</code>	The nominal y-values
<code>sd_y</code>	Standard deviation of the y-values
<code>angle</code>	Angle for the line ending the errorbars (if a value other than 90 is chosen, it makes arrows)
<code>code</code>	Which errorbars should be drawn: <code>code=1</code> means only the lower errorbars (at <code>y-sd_y</code> ), <code>code=2</code> means only the upper errorbars (at <code>y+sd_y</code> ), <code>code=3</code> means both errorbars (from <code>y-sd_y</code> to <code>y+sd_y</code> )
<code>horiz</code>	Logical flag indicating whether horizontal errobars should be drawn horizontal instead of vertical. They will be centered on the same points, but extending horizontally instead of vertically
<code>...</code>	Additional graphical parameters to pass down to the underlying <a href="#">arrows</a> function

**Author(s)**

Thomas Braschler

---

<code>get_associated_values</code>	<i>get_associated_values</i>
------------------------------------	------------------------------

---

**Description**

Looks up corresponding lines in a reference table

**Usage**

```
get_associated_values(descriptive_data,lookup_data,lookup_name_correspondence=NULL,FUN=NULL,look
```

**Arguments**

<code>descriptive_data</code>	Base table indicating the measurements for which a reference value should be found. Is typically a dataframe or matrix
<code>lookup_data</code>	Reference table in which the lookup should be carried out . Is typically a dataframe or matrix; usually, it will have different dimensions than the <code>descriptive_data</code> table
<code>lookup_name_correspondence</code>	Translation table for column correspondence in <code>descriptive_data</code> and <code>lookup_data</code> . Should be a matrix of two columns, the first column containing the column names or indices that should be taken into account for <code>descriptive_data</code> , the second column being the corresponding column names or indices for <code>lookup_name_correspondence</code>
<code>FUN</code>	Aggregation function. Based on the translation table <code>lookup_name_correspondence</code> , <code>get_associated_values</code> will run row by row through <code>descriptive_data</code> , trying to find matching rows in <code>lookup_name_correspondence</code> based on the translation indicated in <code>lookup_name_correspondence</code> . For each of these lines, the element designated by <code>lookup_value_col</code> is selected, and <code>FUN</code> is applied to a vector containing all these elements. Hence, <code>FUN</code> should accept a vector as its input, and return a single value. By default, <code>FUN</code> is chosen to be <code>mean</code> , such that the average value for the matching rows in <code>lookup_data</code> is calculated for every row in <code>descriptive_data</code>
<code>lookup_value_col</code>	Column containing the reference values of interest in <code>lookup_data</code> ; by default, this is assumed to be the last column of <code>lookup_data</code>
<code>...</code>	Additional arguments to be passed on to <code>FUN</code>

**Details**

The function can be thought of as an SQL join between `descriptive_data` and `lookup_data`, the matching condition being that the values in the columns listed in `lookup_name_correspondence` should be identical. The column to be selected in `lookup_data` is given by `lookup_value_col`. As in general, there may be several rows in `lookup_data` matching a given row in `descriptive_data`, the values selected need to be aggregated into a single numerical value; this is performed by `FUN`

**Value**

Of vector with length equal to the number of rows in `descriptive_data`

**Author(s)**

Thomas Braschler

**Examples**

```
descriptive = data.frame(matching_col1=c(1,1,2,2,3,3), matching_col2=c(16,17,16,17,16,17), irrelevant_col1=c(1,1,1,1,1,1))
lookup=data.frame(match_col_1=c(1,1,1,2,2,3,3,3), match_col_2=c(16,17,17,16,17,17,16,17,17), value_col=c(1,1,1,1,1,1,1,1,1))
lookup_name_correspondence=matrix(data=c("matching_col1", "match_col_1", "matching_col2", "match_col_2"), ncol=4, byrow=TRUE)
lookup_value_col="value_col"
assoc_vals=get_associated_values(descriptive_data=descriptive, lookup_data=lookup, lookup_name_correspondence=lookup_name_correspondence, FUN=mean)
descriptive_and_associated=descriptive
descriptive_and_associated$associated = assoc_vals
```

```

cat("The base data: ")
descriptive
cat("The lookup table:")
lookup
cat("The associated values found (here, the sum of corresponding lines):")
assoc_vals
cat("The associated values in comparison with the base data:")
descriptive_and_associated

```

---

get_t_test_matrix	<i>Get a T-Test matrix comparing the results under different treatments</i>
-------------------	---

---

## Description

Compares the data obtained for each value of the controlling factor, and constructs a matrix with the t-test results

## Usage

```
get_t_test_matrix(treatment_factor, data, ...)
```

## Arguments

treatment_factor	Factor indicating the treatment conditions
data	Data obtained under the different conditions
...	Additional arguments to be passed down to the underlying <code>t.test</code> function

## Details

The full list of data should be given, so that standard deviations can be calculated. More precisely, there should be several entries where `treatment_factor` has identical values.

## Value

A matrix containing the p-values of t-tests, comparing the data for each of the possible pairwise combinations of `treatment_factor`

## Author(s)

Thomas Braschler

## Examples

```

test_data=data.frame(condition=c(rep("A",5),rep("B",5),rep("C",8)),outcome=c(1,3,2,2.5,3.2,8,8.25,9,8.5,7.5,
get_t_test_matrix(test_data$condition,test_data$outcome)

```



---

multipleHistograms	<i>multipleHistograms</i>
--------------------	---------------------------

---

## Description

Draws several histograms on the plot area. They are arranged vertically, and share a common x-axis. This a convenience function, it calls several [hist](#) for each group as defined by the argument `levels`

## Usage

```
multipleHistograms(x, levels, breaks = "Sturges", overhead = 0.2, xlab = "", category_labels = NULL, add.legend = FALSE, legend.args = vector(mode = "list", length = 0), barcolors = NULL, cex.axis = 1, FUN = NULL, ...)
```

## Arguments

<code>x</code>	Observations for the counts, as for <a href="#">hist</a> . Vector of numerical values.
<code>levels</code>	Assignment of the observations to the different histograms. Must be a vector of the same length as <code>x</code> ; for each unique value in <code>levels</code> , a histogram will be drawn via a call to <a href="#">hist</a> for the counts and <a href="#">barplot</a> for the actual drawing
<code>breaks</code>	Algorithm or direct indication of the breaks for the histogram counts as for <a href="#">hist</a>
<code>overhead</code>	Graphical parameter for the spacing between the figures. The larger, the more stuffed the page will appear
<code>xlab</code>	Label for the x-axis as for general plotting functions (cf. <a href="#">plot</a> )
<code>category_labels</code>	Labels to be displayed along the bottom-most histogram
<code>add.legend</code>	Whether or not a legend should be added
<code>legend.text</code>	Text for legend. Corresponds to the <code>legend</code> argument of the function <a href="#">legend</a> which is called for adding the legend
<code>legend.args</code>	Additional arguments to be passed down to <a href="#">legend</a> . Must be a named list
<code>barcolors</code>	Colors for the histogram. Should be a vector with as many elements as there are unique values in <code>levels</code> . The colors are hexadecimal values of the type "#808AC0", with RGB coding
<code>cex.axis</code>	Character expansion for the axis; character expansion in R basically means the size of some text, here it is the size of the category labels appearing under the axis)
<code>FUN</code>	Call back function. If provided, will be called for each histogram. The idea is that in this fashion, additional things can be drawn on a histogram. This function must take three arguments: <code>FUN(level, histogram_info, x_vals)</code> . The first argument, <code>level</code> , indicates the level associated with the current histogram; <code>histogram_info</code> gives the information concerning the current counts (corresponds to the return of <a href="#">hist</a> ); <code>x_vals</code> gives the local x-coordinates associated with the center of the bars for the current histogram (e.g., this is the result of the local calls of <a href="#">barplot</a> )
<code>...</code>	Additional graphical parameters to pass down to the underlying <a href="#">barplot</a> function, which does the actual drawing

**Author(s)**

Thomas Braschler

---

pie_with_errorbars	<i>pie_with_errorbars</i>
--------------------	---------------------------

---

**Description**

Plots a pie chart with the option of placing errorbars

**Usage**

```
pie_with_errorbars(x, sd_x=NULL, labels = names(x), edges = 200, radius = 0.8, clockwise = FALSE,
  init.angle = if (clockwise) 90 else 0, density = NULL, angle = 45,
  col = NULL, border = NULL, lty = NULL, main = NULL, initiate_plot=TRUE, col_errorbar=NULL,...)
```

**Arguments**

x	As in <a href="#">pie</a> : A vector of non-negative numerical quantities. The values in x are displayed as the areas of pie slices
sd_x	Errorbars to be displayed for each slice.
labels	As in <a href="#">pie</a> : one or more expressions or character strings giving names for the slices.
edges	As in <a href="#">pie</a> : the circular outline of the pie is approximated by a polygon with this many edges.
radius	As in <a href="#">pie</a> : the pie is drawn centered in a square box whose sides range from -1 to 1. If the character strings labeling the slices are long it may be necessary to use a smaller radius.
clockwise	As in <a href="#">pie</a> : logical indicating if slices are drawn clockwise or counter clockwise (i.e., mathematically positive direction), the latter is default.
init.angle	As in <a href="#">pie</a> : number specifying the starting angle (in degrees) for the slices. Defaults to 0 (i.e., 3 o'clock) unless clockwise is true where init.angle defaults to 90 (degrees), (i.e., 12 o'clock).
density	As in <a href="#">pie</a> : the density of shading lines, in lines per inch. The default value of NULL means that no shading lines are drawn. Non-positive values of density also inhibit the drawing of shading lines.
angle	As in <a href="#">pie</a> : the slope of shading lines, given as an angle in degrees (counter-clockwise).
col	As in <a href="#">pie</a> : a vector of colors to be used in filling or shading the slices. If missing a set of 6 pastel colours is used, unless density is specified when par("fg") is used.
border	As in <a href="#">pie</a> : (possibly vector) argument passed to <a href="#">polygon</a> which draws each slice.
lty	As in <a href="#">pie</a> : (possibly vector) argument passed to <a href="#">polygon</a> which draws each slice.
main	As in <a href="#">pie</a> : an overall title for the plot.

<code>initiate_plot</code>	If true, a new plot is drawn, otherwise the function draws onto the currently active plot
<code>col_errorbar</code>	As in <a href="#">pie</a> : Colors for the errorbars
<code>...</code>	All additional arguments have the same meaning as for the underlying <a href="#">pie</a> function.

**Author(s)**

Thomas Braschler

**Examples**

```
pie_with_errorbars(x=c(1,2,3),sd_x=c(0.1,0.2,0))
```

---

<code>plot_counts</code>	<i>plot_counts</i>
--------------------------	--------------------

---

**Description**

Plots count data with errorbars, significance levels and regression model

**Usage**

```
plot_counts(x,y,sd_y=NULL,sig_codes=NULL,showlinreg="NONE", plot.new = TRUE, groups=NULL, group_order=NULL, weights=NULL)
```

**Arguments**

<code>x</code>	x values
<code>y</code>	associated y values
<code>sd_y</code>	standard deviation of the y values
<code>sig_codes</code>	Codes to indicate significances levels above each data point
<code>showlinreg</code>	Indicates what type of regression line should be shown. Options are "LINEAR", "SQUARE", and "NONE"
<code>plot.new</code>	If true, a new plot is drawn, if false, the line is added to the existing active plot
<code>groups</code>	Vector of the same length as x, y and sd_y, indicates to which group the elements belong. Leave out if there the values are not grouped
<code>group_order</code>	Indicating the order in which the groups should be plotted. This is important for the coloring of the lines. If no <code>col</code> argument is provided in the optional argument, the standard colors available via <code>palette()</code> will be used, otherwise the colors indicated with the "col" argument are used. The <code>col</code> argument should then be a vector of the same length as the <code>group_order</code> argument
<code>weights</code>	Explicit specification of the weights for the linear regression (useful only if a linear regression model is chosen). By default, the weights are taken to be inversely proportional to the standard deviation of the data points.

**category\_bounds**

If supplied, the points are grouped into categories, using the x values and the `category_bounds` to assign the y values. For each category, the mean y value is then displayed at the category midpoint x value, along with errorbars as calculated by [sd\\_mean](#)

**sig.cex**

Character expansion (cex) for the significance labels. Relevant only if `sig_codes` is supplied

**sd\_x**

Standard deviation of x values if available

**...**

Additional graphical parameters, such as xlab, ylab, col

**Details**

The function is intended to handle an x vector with ascending, unique values, and associated y data

**Author(s)**

Thomas Braschler

**Examples**

```
plot_counts(x=c(1,2,3),y=c(2,2,3),sd_y=c(1,1,0.5))
```

---

plot\_dot\_column

*plot\_dot\_column*

---

**Description**

Dot-plot showing individual values, with lateral shift if the y values are too close to show all the values

**Usage**

```
plot_dot_column(x,y,y_threshold_for_shifting=0.1,lateral_shift=0.06,type="p",...)
```

**Arguments**

**x** x values

**y** associated y values

**y\_threshold\_for\_shifting**

Do lateral shift for display if two y-values with the same x-value differ by less than the threshold value

**lateral\_shift** Lateral shift to be used between the points

**type** Type argument to be passed to [lines](#) used internally

**...** Additional arguments to be passed to [lines](#) used internally

**Details**

The function is intended to handle an x vector with ascending, unique values, and associated y data

**Author(s)**

Thomas Braschler

**Examples**

```
plot_dot_column(x=c(1,2,3),y=c(2,2,3),sd_y=c(1,1,0.5))
```

---

sd_mean	<i>sd_mean</i>
---------	----------------

---

**Description**

Calculates the standard deviation of the mean

**Usage**

```
sd_mean(x, na.rm = FALSE)
```

**Arguments**

x	Data vector
na.rm	If true, NA values are removed prior to calculation

**Details**

The standard deviation of the mean is the standard deviation per measurement divided by the square root of the number of measurements

**Value**

Estimated standard deviation of the mean

**Author(s)**

Thomas Braschler

**Examples**

```
x<-c(1,2,3,4,3.5,2.5,1)
sd(x)
sd_mean(x)
```

---

significance_labels	<i>significance_labels</i>
---------------------	----------------------------

---

**Description**

Converts a numerical p-value matrix or vector into a corresponding character matrix or vector with the text labels for significance

**Usage**

```
significance_labels(x, levels=c(0.1, 0.05, 0.01, 0.001), codes=c(".", "*", "**", "***"))
```

**Arguments**

<code>x</code>	P-value vector or matrix
<code>levels</code>	Cut-off levels for significance
<code>codes</code>	Significance codes for values below the corresponding levels

**Details**

`levels` should be sorted in decreasing order, and the `codes` indicate the code to show for values below or equal to the corresponding level

**Value**

A matrix or vector of dimensions identical to `x`, but containing the text codes for the significance level rather than the numerical p-values.

**Author(s)**

Thomas Braschler

**Examples**

```
significance_labels(c(0.01, 0.04, 0.05, 0.5))
```

# Index

- \* **misc**
  - barplot.dots, 2
  - barplot.with.errorbars, 3
  - barplot.xpos, 4
  - change.diagram, 5
  - errorbars, 6
  - get.associated.values, 6
  - get.t.test.matrix, 8
  - multipleHistograms, 9
  - pie.with.errorbars, 10
  - plot.counts, 11
  - plot.dot.column, 12
  - sd.mean, 13
  - significance.labels, 14
- \* **package**
  - plot.counts-package, 2

arrows, 4, 6

axis, 5

barplot, 4, 9

barplot.dots, 2

barplot.with.errorbars, 3, 3

barplot.xpos, 4

change.diagram, 5

errorbars, 6

get.associated.values, 6

get.t.test.matrix, 8

hist, 9

legend, 9

lines, 5, 12

multipleHistograms, 9

par, 2, 3, 5

pie, 10, 11

pie.with.errorbars, 10

plot, 5, 9

plot.counts (plot.counts-package), 2

plot.counts-package, 2

plot.window, 5

plot.counts, 11

plot.dot.column, 3, 12

polygon, 10

sd.mean, 12, 13

significance.labels, 14

t.test, 8

title, 5