

Package: `sciplot` (via `r-universe`)

August 23, 2024

Version 1.2-0

Title Scientific Graphing Functions for Factorial Designs

Author Manuel Morales <mmorales@williams.edu>, with code developed by the R Development Core Team and with general advice from the R-help listserv community, especially Duncan Murdoch.

Maintainer Manuel Morales <Manuel.A.Morales@williams.edu>

Depends stats

Description A collection of functions that creates graphs with error bars for data collected from one-way or higher factorial designs.

License GPL (>= 2)

NeedsCompilation no

Repository <https://mutualism.r-universe.dev>

RemoteUrl <https://github.com/mutualism/sciplot>

RemoteRef HEAD

RemoteSha cac1ce812e004ce663cdffb1c97cf29220e64464

Contents

<code>bargraph.CI</code>	2
<code>lineplot.CI</code>	4
<code>se</code>	6

Index	7
--------------	---

bargraph.CI

*Bar Graph of Calculated Means and CI***Description**

Barplot of the mean and standard error (or other summary statistics) of a response variable for one-way or higher experimental designs.

Usage

```
bargraph.CI(x.factor, response, group=NULL, split=FALSE,
            xlab=NULL, ylab=NULL, col=NULL, angle=NULL, density=NULL,
            lc=TRUE, uc=TRUE, legend=FALSE, ncol=1,
            leg.lab=NULL, x.leg=NULL, y.leg=NULL, cex.leg=1,
            bty="n", bg="white", space=if(split) c(-1,1),
            err.width=if(length(levels(as.factor(x.factor)))>10) 0 else .1,
            err.col="black", err.lty=1,
            fun = function(x) mean(x, na.rm=TRUE),
            ci.fun= function(x) c(fun(x)-se(x), fun(x)+se(x)),
            ylim=NULL, xpd=FALSE, data=NULL, subset=NULL, ...)
```

Arguments

x.factor	a factor (required) whose levels will form the x axis.
response	a numeric variable giving the response.
group	grouping factor (optional) whose levels will form groups of bars for each level of x.factor.
split	logical. Should groups of bars be displayed back to back. See details below.
xlab	default x-axis label. If not specified will parse name from x.factor. To leave blank set to NA
ylab	default y-axis label. If not specified will parse name from response. To leave blank set to NA
col	default color(s) for bars.
angle	default angle of shading lines.
density	default density of shading lines.
lc, uc	logical. Should upper or lower CI be drawn?
legend	logical. Should a legend be included?
ncol	number of columns to use for legend.
leg.lab	legend labels for trace factors.
x.leg, y.leg	optional values to over ride the default legend placement.
cex.leg	character expansion value for legend labels.

bty	line type of bounding box for legend. Defaults to 'none'.
space	set spacing for bars.
err.width	set width of whiskers for error bars.
err.col	color for error bars. Defaults to "black".
err.lty	line type for error bars.
bg	background color of legend.
fun	the function to compute the summary statistic. Should return a single real value. Defaults to mean with NA values removed.
ci.fun	the functions to compute the CI. Should return a vector of length 2 defining the lower and upper limits of the CI's. Defaults to the mean +/- 1 standard error, with NA values removed.
ylim	range of y axis.
xpd	defaults to xpd=FALSE (in contrast to barplot) to cut off the range of data plotted.
data	an optional data frame.
subset	an optional expression indicating the subset of the rows of 'data' that should be used in the plot.
...	further graphical parameters. See barplot() for additional options.

Details

Plots a response as a function of treatment (factor) combinations for one-way and higher designs. This is a "wrapper" function for barplot that adds confidence intervals - barplot should be consulted for details. Note that the option "beside=TRUE" from barplot is hardcoded and that the default for value xpd is FALSE. If the option split is specified, the response associated with the second grouping factor will be made negative and displayed back-to-back with the first.

See Also

barplot, [se](#)

Examples

```
data(ToothGrowth)

# One way design
bargraph.CI(x.factor = dose, response = len, data = ToothGrowth)

# Two-way design with options
bargraph.CI(dose, len, group = supp, data = ToothGrowth,
            xlab = "Dose", ylab = "Growth", cex.lab = 1.5, x.legend = 1,
            col = "black", angle = 45, cex.names = 1.25,
            density = c(0,20), legend = TRUE)
```

lineplot.CI

*Line Plot with CI***Description**

Lineplot of the mean and standard error (or other summary statistics) of a response variable for one-way or higher experimental designs.

Usage

```
lineplot.CI(x.factor, response, group=NULL, type="b", xlab=NULL, ylab=NULL,
  x.cont=FALSE, legend=TRUE, leg.lab=NULL, fixed=FALSE,
  x.leg=NULL, y.leg=NULL, cex.leg=1, ncol=1,
  pch=c(16, 21, 15, 22, 17, 24, c(3:14)),
  fun = function(x) mean(x, na.rm=TRUE),
  ci.fun= function(x) c(fun(x)-se(x), fun(x)+se(x)),
  err.width = if(length(levels(as.factor(x.factor))) > 10) 0 else 0.1,
  err.col = col, err.lty = 1,
  xlim=NULL, ylim=NULL, cex=NULL, lwd=NULL, col="black", cex.axis=1,
  xaxt="s", data=NULL, subset=NULL, ...)
```

Arguments

x.factor	a factor (required) whose levels will form the x axis.
response	a numeric variable giving the response.
group	grouping factor (optional) whose levels will form the traces.
type	the type of plot: lines, points, or both. Defaults to both.
xlab	default x-axis label. If not specified will parse name from x.factor. To leave blank set to NA
ylab	default y-axis label. If not specified will parse name from response. To leave blank set to NA
x.cont	logical. Treat x.factor as a continuous variable?
legend	logical. Should a legend be included?
leg.lab	legend labels for trace factors.
fixed	logical. Should the legend be in the order of the levels of 'trace.factor' or in the order of the traces at their right-hand ends?
x.leg, y.leg	optional values to over ride the default legend placement.
cex.leg	character expansion value for legend labels.
ncol	number of columns to use for legend.
pch	a vector of plotting symbols or characters.
fun	the function to compute the summary statistic. Should return a single real value. Defaults to mean with NA values removed.

<code>ci.fun</code>	the functions to compute the CI. Should return a vector of length 2 defining the lower and upper limits of the CI's. Defaults to the mean +/- 1 standard error, with NA values removed.
<code>err.width</code>	set width of whiskers for error bars.
<code>err.col</code>	color for error bars. Defaults to <code>col</code> .
<code>err.lty</code>	line type for error bars.
<code>xlim, ylim</code>	range for x and y axes.
<code>cex</code>	overall plot character expansion value.
<code>lwd</code>	determines line width.
<code>col</code>	default color(s) for plot.
<code>cex.axis</code>	character expansion value for axis labels.
<code>xaxt</code>	should x-axis be drawn?
<code>data</code>	an optional data frame.
<code>subset</code>	an optional expression indicating the subset of the rows of 'data' that should be used in the plot.
<code>...</code>	further graphical parameters.

Details

Plots a response as a function of treatment (factor) combinations for one-way and higher designs. This is a "wrapper" function for `plot` in one-way designs and `interaction.plot` in higher-way designs that adds confidence intervals - those functions should be consulted for details. Notable changes from the defaults for `interaction.plot` include the removal of NA values by default, options to specify a subset of data and greater flexibility in the placement of legends. This function replicates some of the functionality of `plotmeans` from the package `gplots`, with differences in the treatment of two-way and higher designs.

See Also

`interaction.plot`, [se](#)

Examples

```
data(ToothGrowth)

# One way design
lineplot.CI(x.factor = dose, response = len, data = ToothGrowth)

# Two-way design with options
lineplot.CI(dose, len, group = supp, data = ToothGrowth, cex = 2,
            xlab = "Dose", ylab = "Growth", cex.lab = 1.5, x.legend = 1,
            col = c("blue", "red"), pch = c(16, 16))
```

`se`*Standard Error*

Description

This function computes the standard error of the values in a vector 'x'. If 'na.rm' is 'TRUE' then missing values are removed before computation proceeds.

Usage

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

Arguments

`x` a numeric vector.
`na.rm` logical. Should missing values be removed?

See Also

`sd`

Examples

```
se(1:10) ^ 2
```

Index

* **hplot**

 bargraph.CI, 2

 lineplot.CI, 4

* **univar**

 se, 6

bargraph.CI, 2

lineplot.CI, 4

se, 3, 5, 6