

Package: epade (via r-universe)

October 14, 2024

Type Package

Title Easy Plots

Version 0.5.1

Date 2022-10-25

Depends plotrix, R (>= 4.0.0)

Imports stats, survival, Hmisc

Author Andreas Schulz

Maintainer Andreas Schulz <ades-s@web.de>

Description A collection of nice plotting functions directly from a data.frame with limited customisation possibilities.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2022-10-27 15:35:16 UTC

Contents

epade-package	2
bar.plot.ade	3
bar.plot.wtd	5
bar3d.ade	8
bland.altman.ade	9
box.plot.ade	11
box.plot.wtd	13
correlogram.ade	15
curves.ade	16
format_n.ade	18
format_p.ade	19
histogram.ade	20
KM.plot.ade	21

kurtosis.ade	23
missiogram.ade	24
parallel.ade	25
parallel.set.ade	26
performance.plot.ade	27
qq.ade	29
ratio.plot.ade	30
roc.plot.ade	32
round_n.ade	33
scatter.ade	34
skewness.ade	36
tornado.ade	37
wall.ade	38

Index	41
--------------	-----------

epade-package	<i>Easy Plots</i>
---------------	-------------------

Description

This package is a collection of statistical plots. They are aimed at making fast overview plots from a data frame without elaborate preparations of data. It is my first R package. The main motivation for making it was to learn R. At the moment there is no error protection. Be careful if you use any of the statistical tests in the plots, since they do not necessarily make sense.

Details

Package:	epade
Title:	Easy Plots
Type:	Package
Version:	0.5.1
Date:	2022-10-25
Depends:	plotrix, R (>= 2.12)
Suggests:	survival
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Andreas Schulz
 Maintainer: <ades-s@web.de>

bar.plot.ade	<i>Bar plot</i>
--------------	-----------------

Description

A function to draw a barplot

Usage

```
bar.plot.ade(x, y=NULL, z=NULL, data=NULL,
            vnames.x=NULL, vnames.y=NULL, vnames.z=NULL,
            btext=NULL, b=NULL, b2=0.5, v=NULL, h=NULL, gradient=FALSE,
            xlab="", ylab="", main="", ylim=NULL,
            yticks=NULL, col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL,
            alpha=NULL, beside=TRUE, legendon="topright", wall=0,
            lhoriz=NULL, prozent=FALSE, ploc=0, form="r", border=TRUE,
            density=NULL, angle=NULL, density2=NULL, angle2=NULL, fill=NULL,
            lwd=1, lty=1, blwd=1, b1ty=1)
```

Arguments

x	<ul style="list-style-type: none"> • a factor • a string with the name of the factor variable in the data.frame • a formula x~y or x~y+z • a table
y	<ul style="list-style-type: none"> • second factor • a string with the name of second factor in the data.frame
z	<ul style="list-style-type: none"> • third factor • a string with the name of thirds factor in the data.frame
data	a data.frame
vnames.x	a vector of character strings with labels for the levels of x
vnames.y	a vector of character strings with labels for the levels of y
vnames.z	a vector of character strings with labels for the levels of z
btext	<ul style="list-style-type: none"> • logical asking whether to draw p-values from chisq test • a vector of character strings with test to draw over the bars
b	width of bars in [0, 1]
b2	depth of 3d bars in [0, 1]
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
gradient	logical asking whether to draw a color gradient
xlab	a title for the x axis
ylab	a title for the y axis

main	an overall title for the plot
ylim	the y limits (y1, y2) of the plot
yticks	the number of ticks on the y axis
col	color for the bars representing levels of y
tcol	color of the text in whole plot
bgcol	the background color for plot dekokoration
lcol	colors for the lines to shading bars, a vector is possible
alpha	a parameter in [0, 1] for semi-transparency of bars
beside	logical asking whether to draw bars beside or on top
legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft" • "top" • "topright" • "right" • "center" • "none" <p>This places the legend on the inside of the plot frame at the given location. To locate 2 legends you can give a vector of 2 keywords.</p>
wall	a number between 0 and 6 for selection the dekokoration style of the plot.
lhoriz	logical asking whether to draw legend horizontal
prozent	logical asking whether to draw percents on bars
ploc	Position of percents <ul style="list-style-type: none"> • 0: middle • 1, bottom • 2: over • 3: top • 4: under
form	a single keyword from: <ul style="list-style-type: none"> • 'r': Rects • 'c', 3D Rects • 'z': Zylinders (not working well)
border	logical asking whether to draw borders os bars
density	first density for shading lines, in lines per inch.
angle	first angle (in degrees) for the shading lines.
density2	second density for shading lines, in lines per inch.

angle2	second angle (in degrees) for the shading lines.
fill	fill color for bars if used density, because the col parameter will be used for color of the shading lines.
lwd	width for shading lines
lty	linetype for shading lines
blwd	width for bar-borders
blty	linetype for bar-borders

See Also

[bar3d.ade](#)

Examples

```
x<- round(runif(1000, 0.5, 10.5))
bar.plot.ade(x, btext='Uniform distribution', gradient=TRUE)
x<-rbinom(1000, 1, 0.75)
y<-rbinom(1000, 1, 0.30)
z<-rbinom(1000, 1, 0.50)
bar.plot.ade(x,y,z)
bar.plot.ade(x,y,z, wall=4, form='c', main='Bar-Plot')
```

bar.plot.wtd

weighted Bar plot

Description

A function to draw a weighted or unweighted barplot

Usage

```
bar.plot.wtd(x, y=NULL, z=NULL, w=NULL, data=NULL,
             vnames.x=NULL, vnames.y=NULL, vnames.z=NULL,
             btext=NULL, cutz=F, zperc=NULL,
             b=NULL, b2=0.5, v=NULL, h=NULL, gradient=FALSE,
             xlab="", ylab="", main="", ylim=NULL,
             yticks=NULL, col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL,
             alpha=NULL, beside=TRUE, legendon="topright", wall=0,
             lhoriz=NULL, prozent=FALSE, ploc=0, form="r", border=TRUE,
             density=NULL, angle=NULL, density2=NULL, angle2=NULL, fill=NULL,
             lwd=1, lty=1, blwd=1, blty=1)
```

Arguments

x	<ul style="list-style-type: none"> • a factor • a string with the name of the factor variable in the data.frame • a formula $x\sim y$ or $x\sim y+z$ • a table
y	<ul style="list-style-type: none"> • second factor • a string with the name of second factor in the data.frame
z	<ul style="list-style-type: none"> • third factor • a string with the name of thirds factor in the data.frame
w	<ul style="list-style-type: none"> • a vector of weights • a string with the name of weight variable in the data.frame
data	a data.frame
vnames.x	a vector of character strings with labels for the levels of x
vnames.y	a vector of character strings with labels for the levels of y
vnames.z	a vector of character strings with labels for the levels of z
btext	<ul style="list-style-type: none"> • logical asking whether to draw p-values from chisq test • a vector of character strings with test to draw over the bars
cutz	logical asking whether to use z variable to split bars or to calculate prozent of positive only.
zperc	<p>a single keyword from:</p> <ul style="list-style-type: none"> • "overall" • "rows" • "cols" • "zells" <p>What percentages from z should be calculated?</p>
b	width of bars in [0, 1]
b2	depth of 3d bars in [0, 1]
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
gradient	logical asking whether to draw a color gradient
xlab	a title for the x axis
ylab	a title for the y axis
main	an overall title for the plot
ylim	the y limits (y1, y2) of the plot
yticks	the number of ticks on the y axis
col	color for the bars representing levels of y
tcol	color of the text in whole plot
bgcol	the background color for plot dekoration

lcol	colors for the lines to shading bars, a vector is possible
alpha	a parameter in [0, 1] for semi-transparency of bars
beside	logical asking whether to draw bars beside or on top
legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft" • "top" • "topright" • "right" • "center" • "none" <p>This places the legend on the inside of the plot frame at the given location. To locate 2 legends you can give a vector of 2 keywords.</p>
wall	a number between 0 and 6 for selection the decoration style of the plot.
lhoriz	logical asking whether to draw legend horizontal
prozent	logical asking whether to draw percents on bars
ploc	Position of percents <ul style="list-style-type: none"> • 0: middle • 1, bottom • 2: over • 3: top • 4: under
form	a single keyword from: <ul style="list-style-type: none"> • 'r': Rects • 'c', 3D Rects • 'z': Zylinders (not working well)
border	logical asking whether to draw borders os bars
density	first density for shading lines, in lines per inch.
angle	first angle (in degrees) for the shading lines.
density2	second density for shading lines, in lines per inch.
angle2	second angle (in degrees) for the shading lines.
fill	fill color for bars if used density, because the col parameter will be used for color of the shading lines.
lwd	width for shading lines
lty	linetype for shading lines
blwd	width for bar-borders
blty	linetype for bar-borders

See Also[bar3d.ade](#)**Examples**

```
x<-rbinom(1000, 1, 0.75)
y<-rbinom(1000, 1, 0.30)
z<-rbinom(1000, 1, 0.50)
w<-abs(rnorm(1000))
bar.plot.wtd(x,y,z, w)
bar.plot.wtd(x,y,z, w, wall=4, form='c', main='Bar-Plot')
```

bar3d.ade

*3D Bar-Plot***Description**

Draw pseudo 3d Bar-Plot

Usage

```
bar3d.ade(x, y=NULL, data=NULL, xw=0.5, zw=1,
          main=NULL, xlab=NULL, ylab=NULL, zlab=NULL,
          xticks=NULL, yticks=NULL, zticks=NULL,
          col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL,
          axes=TRUE, fgbox=TRUE, bgbox=TRUE, wall=0)
```

Arguments

x	<ul style="list-style-type: none"> • a table or matrix • a numeric vector or factor • a string with the name of the variable in the data.frame • a formula x~y
y	<ul style="list-style-type: none"> • a numeric vector or factor • a string with the name of the variable in the data.frame
data	data.frame if used strings of formul for x and y
xw	width of bars in x direction
zw	width of bars in z direction
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
zlab	a title for the z axis
xticks	a vector of labels for the x axis
yticks	the number of ticks on the y axis or a vector of exact ticks

zticks	a vector of labels for the z axis
col	<ul style="list-style-type: none"> • a color for the bars • a vector of colors • a matrix of colors
tcol	color of the text in whole plot
bgcol	the background color for plot dekokation
lcol	bar edges color
alpha	a parameter in [0, 1] for semi-transparency of bars
axes	logical asking whether to plot axis
fgbox	logical asking whether to draw 3d box in forderground (dotted part of box)
bgbox	logical asking whether to draw 3d box (if FALSE, set fgbox to FALSE too)
wall	a number between 0 and 6 for selection the dekokation style of the plot.

See Also

[bar.plot.ade](#)

Examples

```
bar3d.ade(rpois(200,2), rpois(200,2), wall=3)
x <- seq(-16, 16, length= 48)
y <- x
f <- function(x,y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
z <- outer(x, y, f)
z[is.na(z)] <- 1
bar3d.ade(z, wall=2, xw=1, zw=0.2, axes=FALSE, bgbox=FALSE,
xlab='', ylab='', zlab='', alpha=1, col='lavender')
```

bland.altman.ade *Bland-Altman plot*

Description

Plot for assessing agreement between two methods of clinical measurement

Usage

```
bland.altman.ade(x, y, data=NULL, ltext=TRUE, main="Bland-Altman Plot",
xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL,
lwd=2, cex=1, pch=16, lty=c(1,2,2), xticks=NULL, yticks=NULL,
col=NULL, tcol=NULL, bgcol=NULL, lcol=c(4,2,2), alpha=NULL,
fitline=1, wall=0, v=NULL, h=NULL, span=0.75)
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector of first measurement • a string with the name of first measurement in the data.frame
y	<ul style="list-style-type: none"> • a numeric vector of second measurement • a string with the name of second measurement in the data.frame
data	data.frame if used strings for (x,y)
ltext	<ul style="list-style-type: none"> • logical asking whether to draw labels for the lines • a string vector with the labels for the lines
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
lwd	the line width
cex	character (or symbol) expansion: a numerical value
pch	plotting "character", i.e., symbol to use. This can either be a single character or an integer code for one of a set of graphics symbols.
lty	the line type, a vector is possible
xticks	the number of ticks on the x axis or a vector of exact ticks
yticks	the number of ticks on the y axis or a vector of exact ticks
col	color of the points
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the lines in plot, a vector of colors is possible
alpha	a parameter in [0, 1] for semi-transparency of points
fitline	a number between 0 and 3 to fit: <ul style="list-style-type: none"> • 0. not fit • 1. a lm regression line • 2. a loess local regression line • 3. a pylinomial regression line
wall	a number between 0 and 6 for selection the decoration style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
span	the span parameter for lowess curve fit (only if fitline=2)

Details

It is only a Wrapper function for scatter.ade. Plotting the Difference against the mean for both variables.

See Also[scatter.ade](#)**Examples**

```
x<-rnorm(1000, 0, 3)
y<-x+rnorm(1000, 1, 0.5)
bland.altman.ade(x, y, wall=2)
```

box.plot.ade

*Boxplot***Description**

Draw a box, a violin, a box-percentile and more plots for subgroups

Usage

```
box.plot.ade(x, group=NULL, group2=NULL, data=NULL, vnames=NULL,
            main=NULL, xlab=NULL, ylab=NULL, ylim=NULL, yticks=NULL,
            col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, pdigs=4, alpha=NULL,
            cex=1, cex.axis=1, lwd=2, h=NULL, lty=2, test=FALSE,
            varwidth=TRUE, means=FALSE, count=TRUE, zylinder=FALSE,
            twoside=TRUE, paired=FALSE, outlier=TRUE, wall=0, type='box')
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector of values • a character string with the name of the variable in the data.frame • a formula x~group or x~group+group2
group	<ul style="list-style-type: none"> • a factor to group the plots • a character string with the name of the group variable in the data.frame
group2	<ul style="list-style-type: none"> • a second factor to group the plots • a character string with the name of the group2 variable in the data.frame
data	a data.frame
vnames	a vector of character strings with the names of groups in the legend, it can be a list of two vectors if group2 is given
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
ylim	the y limits (y1, y2) of the plot
yticks	the number of ticks on the y axis or a vector of exact ticks
col	color of objects (Boxes)

tcol	color of the text in whole plot
bgcol	the background color for plot dekokation
lcol	color for the lines in plot, a vector of colors is possible
pdigs	a number indicate how to round p-values.: see ?format.pval.ade
alpha	a parameter in [0, 1] for semi-transparency of objects (Boxes)
cex	a numeric character expansion factor for the points
cex.axis	a numeric character expansion factor for axis
lwd	the line width
h	the y-value(s) for horizontal line(s).
lty	the line type
test	<ul style="list-style-type: none"> • logical asking whether to test for the difference or trend between groups • a string to print before p-value. (leads to test=T)
varwidth	logical asking whether the boxwidth indicate the N
means	logical asking whether to draw points for the means
count	<ul style="list-style-type: none"> • logical asking whether to shown N • a string to replace N with it, if you use "?" in you string in will be replaced with N. • a vector of strings to replace N with it, if you use "?" in you strings in will be replaced with N.
zylinder	logical asking whether to draw boxes in a zylinder style
twoside	logical asking whether to perform a wto sided test, or a test with direction
paired	logical indicating whether you want a paired test.
outlier	logical asking whether to draw points for outlier
wall	a number between 0 and 6 for selection the dekokation style of the plot.
type	one of following: <ul style="list-style-type: none"> • "boxplot" for a normal boxplot • "violin" for a violinplot • "sd" for mean and one sd interval • "2sd" for mean and one, two sd intervals • "percentile" for a box-percentile plot • "iqr" for a IQR plot

Details

it test for difference with T-test or U-test depends on the skewness ≤ 1 or >1 . For a trend it perform a Jonckheere-Terpstra trend test.

Examples

```
x<-rnorm(1000)
g<-round(runif(1000))
g2<-round(runif(1000))
box.plot.ade(x, g, g2, vnames=list(c("subgroup 1","subgroup 2"),
c("group 1", "group 2")), wall=0, count='N: ?', means=TRUE)
box.plot.ade(x, g, g2, vnames=list(c("subgroup 1","subgroup 2"),
c("group 1", "group 2")), wall=1, type="violin")
box.plot.ade(x, g, g2, vnames=list(c("subgroup 1","subgroup 2"),
c("group 1", "group 2")), wall=2, type="percentile")
box.plot.ade(x, g, g2, vnames=list(c("subgroup 1","subgroup 2"),
c("group 1", "group 2")), wall=3, type="sd")
```

box.plot.wtd	<i>weighted Boxplot</i>
--------------	-------------------------

Description

Draw a weighted Boxplot. (Beta Version)

Usage

```
box.plot.wtd(x, group=NULL, group2=NULL, w=NULL, data=NULL, vnames=NULL,
main=NULL, xlab=NULL, ylab=NULL, ylim=NULL, yticks=NULL,
col=NULL, tcol=NULL, bcol=NULL, lcol=NULL, pdigs=4, alpha=NULL,
cex=1, cex.axis=1, lwd=2, h=NULL, lty=2,
varwidth=TRUE, means=FALSE, count=TRUE, zylinder=FALSE,
outlier=TRUE, wall=0, type='box')
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector of values • a character string with the name of the variable in the data.frame • a formula $x \sim \text{group}$ or $x \sim \text{group} + \text{group2}$
group	<ul style="list-style-type: none"> • a factor to group the plots • a character string with the name of the group variable in the data.frame
group2	<ul style="list-style-type: none"> • a second factor to group the plots • a character string with the name of the group2 variable in the data.frame
w	weights
data	a data.frame
vnames	a vector of character strings with the names of groups in the legend, it can be a list of two vectors if group2 is given
main	an overall title for the plot
xlab	a title for the x axis

ylab	a title for the y axis
ylim	the y limits (y1, y2) of the plot
yticks	the number of ticks on the y axis or a vector of exact ticks
col	color of objects (Boxes)
tcol	color of the text in whole plot
bgcol	the background color for plot dekokoration
lcol	color for the lines in plot, a vector of colors is possible
pdigs	a number indicate how to round p-values.: see ?format.pval.ade
alpha	a parameter in [0, 1] for semi-transparency of objects (Boxes)
cex	a numeric character expansion factor for the points
cex.axis	a numeric character expansion factor for axis
lwd	the line width
h	the y-value(s) for horizontal line(s).
lty	the line type
varwidth	logical asking whether the boxwidth indicate the N
means	logical asking whether to draw points for the means
count	<ul style="list-style-type: none"> • logical asking whether to shown N • a string to replace N with it, if you use "?" in you string in will be replaced with N. • a vector of strings to replace N with it, if you use "?" in you strings in will be replaced with N.
zylinder	logical asking whether to draw boxes in a zylinder style
outlier	logical asking whether to draw points for outlier
wall	a number between 0 and 6 for selection the dekokoration style of the plot.
type	one of following: <ul style="list-style-type: none"> • "boxplot" for a normal boxplot • "sd" for mean and one sd interval • "2sd" for mean and one, two sd intervals • "iqr" for a IQR plot

Examples

```
x<-rnorm(1000)
g<-round(runif(1000))
w<-abs(rnorm(1000))
d<- data.frame(x, g, w)
box.plot.wtd('x', 'g', w='w', data=d)
```

correlogram.ade	<i>Correlogram plot</i>
-----------------	-------------------------

Description

Draw a correlogram (A Correlation matrix)

Usage

```
correlogram.ade(vars1, vnames1="noname", vars2, vnames2="noname",
  prediktors=0, data=NULL, xlab=NULL, ylab=NULL, main=NULL,
  method="p", digits=2, pdigs=4, pvals=TRUE, bars=TRUE,
  col=NULL, tcol=NULL, bgcol=NULL, wall=0)
```

Arguments

vars1	a vector of character strings with the names of variables in data.frame (rows)
vnames1	a vector of character strings with the labels for vars1
vars2	a vector of character strings with the names of variables in data.frame (cols)
vnames2	a vector of character strings with the labels for vars2
prediktors	a vector of character strings with the names of variables for adjustment (partial correlation)
data	a data.frame
xlab	a title for the x axis
ylab	a title for the y axis
main	a main title of the plot, it will be drawn below the plot
method	a single keyword from: <ul style="list-style-type: none"> • "pearson" • "spearman" • "kendall" indicating which correlation coefficient is to be used.
digits	how many significant digits are to be used
pdigs	a number indicate how to round p-values.: see ?format.pval.ade
pvals	logical asking whether to draw p-values
bars	logical asking whether to draw bars
col	not used
tcol	color of the text in whole plot
bgcol	the background color for plot dekoration
wall	a number between 0 and 6 for selection the dekoration style of the plot.

Examples

```

data<-rnorm(1000)
data<-as.data.frame(data)
for(i in 1:9){
eval(parse(text=paste("data$var_", i, "<- rnorm(1000)", sep='')))
eval(parse(text=paste("data$var_", i,
"round(runif(round(runif(1, 1, 100))), 1, 1000))<-NA", sep='')))
}
correlogram.ade(vars1=c('var_1', 'var_2', 'var_3', 'var_4', 'var_5'),
vars2=c('var_6', 'var_7', 'var_8', 'var_9', 'var_3'),
data=data)

```

curves.ade

Curves

Description

Draw points and a line between the points

Usage

```

curves.ade(x, y=NULL, group=NULL, data=NULL, vnames=NULL,
main=NULL, xlab=NULL, ylab=NULL, legendon="topright",
xlim=NULL, ylim=NULL, lwd=1, lwd2=1, cex=1, pch=16,
lty=1, lty2=2, col=NULL, xticks=NULL, yticks=NULL,
tcol=NULL, bgcol=NULL, alpha=NULL, fitline=0,
wall=0, v=NULL, h=NULL, diag=F, points=T)

```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector of x coordinates for the points • a string with the name of the variable in the data.frame • a formula y~x or y~x+group
y	<ul style="list-style-type: none"> • a numeric vector of y coordinates for the points • a string with the name of the variable in the data.frame • a formula y~x or y~x+group
group	<ul style="list-style-type: none"> • a factor to group the points • a character string with the name of the group variable in the data.frame
data	data.frame if used strings for (x,y,group)
vnames	a vector of character strings with the names of groups in the legend
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
legendon	a single keyword from:

- "bottomright"
- "bottom"
- "bottomleft"
- "left"
- "topleft"
- "top"
- "topright"
- "right"
- "center"

This places the legend on the inside of the plot frame at the given location.

xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
lwd	line width for the lines
lwd2	line width for the fitted lines
cex	character (or symbol) expansion: a numerical value
pch	plotting "character", i.e., symbol to use. This can either be a single character or an integer code for one of a set of graphics symbols.
lty	the line types
lty2	the line type for fitted lines
col	a vector of colors for the lines of each group
xticks	the number of ticks on the x axis or a vector of exact ticks
yticks	the number of ticks on the y axis or a vector of exact ticks
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
alpha	a parameter in [0, 1] for semi-transparency of lines and points
fitline	a number between 0 and 3 to fit: <ul style="list-style-type: none"> • 0. not fit • 1. a lm regression line • 2. a loess local regression line • 3. a polynomial regression line
wall	a number between 0 and 6 for selection the decoration style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
diag	logical asking whether to plot a diagonal line
points	logical asking whether to draw points

Details

It is only a wrapper function for scatter.ade.

See Also[scatter.ade](#)**Examples**

```
x<- -100:100
curves.ade(x, x^2, points=FALSE)
```

format_n.ade	<i>Format a number</i>
--------------	------------------------

Description

Format an R object for pretty printing.

Usage

```
format_n.ade(x, digits=2, scientific=FALSE)
```

Arguments

x	<ul style="list-style-type: none">• a numeric value• a vector of numeric values• a matrix of numeric values
digits	how many significant digits are to be shown
scientific	a logical specifying whether the number should be encoded in scientific format

Value

An object of similar structure to x containing character representations of the elements of x in a common format

See Also[format_p.ade](#)**Examples**

```
format_n.ade((1:10)/100)
```

format_p.ade	<i>Formating p-values</i>
--------------	---------------------------

Description

Format a p-value for pretty printing.

Usage

```
format_p.ade(x, pgits=4, digits=2)
```

Arguments

x	<ul style="list-style-type: none">• a numeric p-value• a vector of numeric p-values• a matrix of numeric p-values
pgits	Number of digits after decimal for cutoff of the p-value, 4 means, all under 0.0001 will be shown like <0.0001
digits	how many significant digits are to be used. (bevore the cut with pgits)

Details

if used a half-integer for pgits, like 3.5 the p-value will be shown like <0.0005

Value

An object of similar structure to x containing character representations of the elements of x in a common format

See Also

[format_n.ade](#)

Examples

```
format_p.ade(0.045825, 4, 3)
format_p.ade(0.000025, 4)
format_p.ade(0.000025, 3.5)
```

 histogram.ade

Histogram

Description

plot multiple histograms in one plot

Usage

```
histogram.ade(x, group=NULL, w=NULL, data=NULL, vnames=NULL, freq=FALSE,
  breaks="Sturges", density=NULL, angle=NULL, xlab=NULL, ylab=NULL,
  main="", xlim=NULL, ylim=NULL, legendon="topright", xticks=NULL,
  col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL, lwd=1,
  kern=TRUE, norm=TRUE, bars=TRUE, wall=0, v=NULL, h=NULL, lty=2)
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector • a string with the name of the variable in the data.frame • a formula x~group
group	<ul style="list-style-type: none"> • a factor to make separate histogram for each class • a string with the name of the group variable in the data.frame
w	weights for weighted histograms
data	a data.frame
vnames	a vector of character strings with the names of groups in the legend
freq	logical: TRUE representation of frequencies or FALSE component density.
breaks	a single number giving the number of cells for the histogram
density	the density of shading lines, in lines per inch. Set it to NA avoid shading lines with wall=0.
angle	the vector of slopes of shading lines, given as an angle in degrees (counter-clockwise).
xlab	a title for the x axis
ylab	a title for the y axis
main	an overall title for the plot
xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left"

	<ul style="list-style-type: none"> • "topleft" • "top" • "topright" • "right" • "center"
	This places the legend on the inside of the plot frame at the given location.
xticks	the number of ticks on the x axis or a vector of exact ticks
col	colors for each histogram
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the lines in plot, a vector of colors is possible, only used if h or v is given
alpha	a parameter in [0, 1] for semi-transparency of histogram
lwd	the line width
kern	logical asking whether to draw lines for kernel density estimation
norm	logical asking whether to draw density from normal distribution for comparison
bars	logical asking whether to draw bars or only polygons of kernel density estimation
wall	a number between 0 and 6 for selection the decoration style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
lty	the line type, a vector of types is possible

Examples

```
g<-rbinom(1000, 1, 0.5)
x<-rnorm(1000)+g
histogram.ade(x, g, wall=3, breaks=24)
histogram.ade(x, g, wall=2, bars=FALSE)
```

KM.plot.ade

Kaplan-Meier curves

Description

plot Kaplan-Meier survival curves

Usage

```
KM.plot.ade(time, event, group=NULL, data=NULL, vnames=NULL,
            main="Kaplan-Meier Plot", xlab="Follow-Up Time",
            ylab="Cumulative Survival", xlim=NULL, ylim=NULL, xticks=NULL,
            legendon='bottomleft', lwd=2, lty=1,
            col=NULL, tcol=NULL, bgcol=NULL, pdigs=4,
            CI=FALSE, ycut=TRUE, zenspoints=FALSE, test=FALSE, wall=0)
```

Arguments

time	<ul style="list-style-type: none"> • a numeric vector for time • a character string with the name of time variable in the data.frame
event	<ul style="list-style-type: none"> • a numeric vector for event (censoring) • a character string with the name of event variable in the data.frame
group	<ul style="list-style-type: none"> • a factor to group the curves • a character string with the name of the group variable in the data.frame
data	data.frame if used character string for (time,event,group)
vnames	a vector of character strings with the names of groups in the legend
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
xticks	the number of ticks on the x axis or a vector of exact ticks
legendon	<p>a single keyword from:</p> <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft" • "top" • "topright" • "right" • "center" <p>This places the legend on the inside of the plot frame at the given location.</p>
lwd	the line width
lty	the line type
col	a vector of colors for each curve
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
pdigs	a number indicate how to round p-values.: see ?format.pval.ade
CI	logical asking whether to plot confidence intervals
ycut	logical asking whether to cut the y axis, if the space is not used
zenspoints	logical asking whether to draw censored datapoint
test	logical asking whether to test for the difference between curves
wall	a number between 0 and 6 for selection the decoration style of the plot.

Details

The p-value comes from a logrank test

Examples

```
times<- sort(abs(rnorm(1000)))
events<- round(runif(1000))
groups<- round(runif(1000, 0, 3))
KM.plot.ade(times, events, groups, wall=2)
```

kurtosis.ade

Simple function to calculate kurtosis

Description

calculate kurtosis

Usage

```
kurtosis.ade(x, na.rm=FALSE)
```

Arguments

x	a numeric vector
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Value

a single number of kurtosis from x

See Also

[skewness.ade](#)

Examples

```
x<-rnorm(1000)
kurtosis.ade(x)
```

missiogram.ade *Missing Value Plot*

Description

Overview of missing values in a data.frame

Usage

```
missiogram.ade(vars=NULL, vnames=NULL, data=NULL, ints=50, nvars=50,
               xlab="ID", ylab="Variables", main="Missing Value Plot",
               ylab2="N. Missings", col=NULL, tcol=NULL, bgcol=NULL, wall=0)
```

Arguments

vars	a vector of character strings with names of variables in data.frame
vnames	a vector of character strings with labels for the variables
data	a data.frame, it is possible to give only the data.frame.
ints	a integer giving number of intervals on x axis
nvars	number of variables in data.frame to be shown if only the data.frame ist given
xlab	a title for the x axis
ylab	a title for the y axis
main	an overall title for the plot
ylab2	a title for the second y axis
col	color of the symbols
tcol	color of the text in whole plot
bgcol	the background color for plot dekokoration
wall	a number between 0 and 6 for selection the dekokoration style of the plot.

Details

One, two or three points indicate respectively number of missing values in this section. More than 3 missing values will be shown with a semi-transparency surface over the section. No semi-transparency means, all the values are missing in this section.

Examples

```
data<-rnorm(1000)
data<-as.data.frame(data)
for(i in 1:20){
  eval(parse(text=paste("data$var_", i, "<- rnorm(1000)", sep='')))
  eval(parse(text=paste("data$var_", i,
    "[round(runif(round(runif(1, 1, 100)), 1, 1000))]<-NA", sep='')))
}
missiogram.ade(data=data)
```

parallel.ade	<i>Parallel coordinate plot</i>
--------------	---------------------------------

Description

Draw a parallel coordinate plot

Usage

```
parallel.ade(vars, vnames=NULL, data=NULL, group=NULL, ylim=NULL,
             xlab=NULL, ylab=NULL, main=NULL,
             alpha=NULL, col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL,
             scale=FALSE, desc=FALSE, means=TRUE,
             legendon="top", wall=0)
```

Arguments

vars	<ul style="list-style-type: none"> • a list of numeric variables • a vector of character strings with the names of variables in data.frame
vnames	a vector of character strings with the names of variables
data	data.frame if used character string for vars
group	<ul style="list-style-type: none"> • a factor to group the lines • a character string with the name of the group variable in the data.frame
ylim	the y limits (y1, y2) of the plot
xlab	a title for the x axis
ylab	a title for the y axis
main	an overall title for the plot
alpha	a parameter in [0, 1] for semi-transparency of points
col	a vector of colors for the lines for each group or each line if not using groups
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the mean lines
scale	a logical specifying whether the variables will be scaled to the range of [0,1]
desc	a logical specifying whether the colors will be sorted decreasingly
means	logical asking whether to draw means
legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft"

- "top"
- "topright"
- "right"
- "center"

This places the legend on the inside of the plot frame at the given location.

wall a number between 0 and 6 for selection the decoration style of the plot.

See Also

[parallel.set.ade](#)

Examples

```
x1<-rnorm(1000, 0, 5)
x2<-rnorm(1000, 0, 4)
x3<-rnorm(1000, 0, 3)
x4<-rnorm(1000, 0, 2)
parallel.ade(vars=list(x1, x2, x3, x4))
g<-rbinom(1000, 1, 0.5)
x1[g==1]<- x1[g==1]+8
x2[g==1]<- x2[g==1]-8
x3[g==1]<- x3[g==1]+6
x4[g==1]<- x4[g==1]-6
parallel.ade(vars=list(x1, x2, x3, x4), group=g,wall=3)
```

parallel.set.ade *Parallel set plot*

Description

Plot proportions of categorical data in parallel manner

Usage

```
parallel.set.ade(vars, vnames=NULL, data=NULL,
                 xlab="Factors", ylab="Proportion", main=NULL,
                 col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL,
                 cex=NULL, wall=0, horizontal=FALSE)
```

Arguments

vars	<ul style="list-style-type: none"> • a list of factors • a vector of character strings with the names of factors in data.frame
vnames	a vector of character strings with the names of factors
data	data.frame if used character string for vars
xlab	a title for the x axis

ylab	a title for the y axis
main	an overall title for the plot
col	a vector of colors for each levels of first factor
tcol	color of the text in whole plot
bgcol	the background color for plot dekokation
lcol	a vector of colors or single color for areas where all levels are drawn
alpha	a parameter in [0, 1] for semi-transparency of polygons
cex	character expansion factor for levels printing
wall	a number between 0 and 6 for selection the dekokation style of the plot.
horizontal	logical asking whether to draw the plot horizontally

See Also

[parallel.ade](#)

Examples

```
x<-rbinom(1000, 1, 0.25)
y<-rbinom(1000, 1, 0.5)
z<-rbinom(1000, 1, 0.75)
parallel.set.ade(list(x,y,z), wall=2)
```

performance.plot.ade *Performance Plot*

Description

Draw for all possible cutoffs, TP, FP, TN, FN, sensitivity, specificity and more.

Usage

```
performance.plot.ade(pred, event, data=NULL, vnames=NULL, cutoffs=NULL,
  cutnames=NULL, main=NULL, xlab="cutoff", ylab="%",
  xlim=NULL, xticks=12,
  col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL,
  nints=100, lty=NULL, lwd=2, stats=c(1, 2),
  youden=TRUE, wall=0)
```

Arguments

pred	<ul style="list-style-type: none"> • a numeric predictor vector • a string with the name of the variable in the data.frame • a formula yevent~pred
event	<ul style="list-style-type: none"> • a numeric event vector

	<ul style="list-style-type: none"> • a string with the name of the variable in the data.frame
<code>data</code>	data.frame if used character string for (pred, event)
<code>vnames</code>	a vector of character strings with the labels for performance values in top legend
<code>cutoffs</code>	a vector of optional cutoffs, to draw vertical lines
<code>cutnames</code>	a vector of character strings with the names for optional cutoffs
<code>main</code>	an overall title for the plot
<code>xlab</code>	a title for the x axis
<code>ylab</code>	a title for the y axis
<code>xlim</code>	the x limits (x1, x2) of the plot
<code>xticks</code>	the number of ticks on the x axis or a vector of exact ticks
<code>col</code>	a vector of 4 colors, for the 4 areas in the plot (TP, TN, FP, FN)
<code>tcol</code>	color of the text in whole plot
<code>bgcol</code>	the background color for plot decoration
<code>lcol</code>	color for the lines in plot, a vector of colors is possible
<code>alpha</code>	a parameter in [0, 1] for semi-transparency of points
<code>nints</code>	number of points for the areas and curves, precision of calculation
<code>lty</code>	a vector of line types
<code>lwd</code>	the line width for all lines, cutoff lines too
<code>stats</code>	<p>a number or vector of numbers from 1 to 4 indicate what statistics are to be drawn</p> <ul style="list-style-type: none"> • 1. Sensitivity • 2. Specificity • 3. PPV • 4. NPV
<code>youden</code>	logical asking whether to plot red line for youden-index maximum
<code>wall</code>	a number between 0 and 6 for selection the decoration style of the plot.

Examples

```
performance.plot.ade(rnorm(100), round(runif(100)))
```

qq.ade	<i>Q-Q Plot</i>
--------	-----------------

Description

Draw a simple Q-Q Plot

Usage

```
qq.ade(x, data=NULL, main="Q-Q Plot",
       xlab="Theoretical Quantiles", ylab="Sample Quantiles",
       xlim=NULL, ylim=NULL, lwd=1, cex=1, pch=16, lty=1,
       xticks=NULL, yticks=NULL,
       col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL, fitline=0,
       qline=TRUE, wall=0, v=NULL, h=NULL, diag=FALSE, band=FALSE, span=0.75)
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector • a character string with the name of the variable in the data.frame
data	data.frame if used character string for x
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
lwd	the line width
cex	character (or symbol) expansion: a numerical value
pch	plotting "character", i.e., symbol to use. This can either be a single character or an integer code for one of a set of graphics symbols.
lty	the line type
xticks	the number of ticks on the x axis or a vector of exact ticks
yticks	the number of ticks on the y axis or a vector of exact ticks
col	a color for the points
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the lines in plot, a vector of colors is possible
alpha	a parameter in [0, 1] for semi-transparency of points
fitline	a number between 0 and 3 to fit: <ul style="list-style-type: none"> • 0. not fit • 1. a lm regression line

	<ul style="list-style-type: none"> • 2. a loess local regression line • 3. a pylinomial regression line
qline	logical asking whether to draw a median line fitted from data between 25th and 75th percentiles only.
wall	a number between 0 and 6 for selection the dekokation style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
diag	logical asking whether to plot a diagonal line
band	logical asking whether to plot a simulated normal band or N of iteration for band estimation
span	the span parameter for lowess curve fit (only if fitline=2)

Details

It is only a wrapper function for scatter.ade.

See Also

[scatter.ade](#)

Examples

```
qq.ade(rnorm(1000))
qq.ade(rchisq(1000, 2), fitline=2, wall=3, col=2)
```

ratio.plot.ade

Ratio plot

Description

A Plot for varying kinds of estimators with intervals

Usage

```
ratio.plot.ade(M, vnames, sectext=NULL,
               main=NULL, xlab=NULL, ylab=NULL, legenlab=NULL, rlab=NULL,
               col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL,
               r=NULL, v=c(0,1), lty=c(1,2), xticks=18,
               hlines=TRUE, legends=TRUE, logaxe=FALSE, wall=0)
```

Arguments

M	A matrix or a list of matrices where first column is the estimated value, second and third are lower and upper interval limits , rows are different values for comparison.
vnames	a vector of character strings with the names for different values (rows)
sectext	a second text to be placed under the vnames, can be p-values for example
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
legenlab	a vector of character strings with the names for groups in the legend
r1ab	a title over the vnames
col	a vector of colors for the intervals in each group
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the lines in plot, a vector of colors is possible
r	a value in (0,1), define the right space for labels.
v	the x-value(s) for vertical line(s).
lty	the line type
xticks	the number of ticks on the x axis or a vector of exact ticks
hlines	logical asking whether to draw horizontal grid lines
legends	logical asking whether to draw the legend
logaxe	logical asking whether to use a logarithmic scale on x axis
wall	a number between 0 and 6 for selection the decoration style of the plot.

Examples

```

vnames<-c('Value 1', 'Value 2', 'Value 3', 'Value 4')
x<-abs(rnorm(4))
M1<-cbind(x, x-(x/2), x+(x/2))
x<-abs(rnorm(4))
M2<-cbind(x, x-(x/2), x+(x/2))
x<-abs(rnorm(4))
M3<-cbind(x, x-(x/2), x+(x/2))
ratio.plot.ade(list(M1, M2, M3), vnames=vnames, wall=3,
legenlab=c('group 1', 'group 2', 'group 3'))

```

roc.plot.ade

*ROC-curves plot***Description**

Function to plot ROC curves with AUC calculation

Usage

```
roc.plot.ade(pred, event, group=NULL, data=NULL, vnames=NULL,
             main="", xlab="1-Specificity", ylab="Sensitivity",
             digits=3, pdigs=4, lty=1, lwd=2,
             col=NULL, tcol=NULL, bgcol=NULL,
             wall=0, test=FALSE, CC=TRUE, auc=TRUE, diag=TRUE, spec=FALSE)
```

Arguments

pred	<ul style="list-style-type: none"> • a list of numeric predictor variables • a vector of character strings with the names of the predictors in data.frame
event	<ul style="list-style-type: none"> • a numeric event variable • a character strings with the names of event variable in data.frame
group	<ul style="list-style-type: none"> • a factor to group the curves • a character strings with the names of factor variable in data.frame
data	data.frame if used character string for (pred,event,group)
vnames	a vector of character strings with the names of groups in the legend
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
digits	how many significant digits are to be shown for AUC
pdigs	a number indicate how to round p-values.: see ?format.pval.ade
lty	a single line type or a vector og line types
lwd	the line width
col	a vector of colors for each curve
tcol	color of the text in whole plot
bgcol	the background color for plot dekokoration
wall	a number between 0 and 6 for selection the dekokoration style of the plot.
test	logical asking whether to test for the difference between curves
CC	logical asking whether to use complete cases for all curves
auc	logical asking whether to draw AUC in legend
diag	logical asking whether to plot a diagonal line
spec	logical asking whether to draw a axis for Specificity at top.

Details

if test is TRUE the function perform a DeLong-DeLong test for correlated ROC-curves

Examples

```
# simple curve
event<-rbinom(1000, size=1, prob=0.3)
pred <- event+rnorm(1000)
roc.plot.ade(pred, event)
# grouped
group=rbinom(1000, 1 ,0.5)
roc.plot.ade(pred, event, group, wall=2)
# comparison of two predictors
pred2 <- event+rnorm(1000, 0, 2)
roc.plot.ade(list(pred, pred2), event, test=TRUE, wall=3)
```

round_n.ade

A round function

Description

round a numeric value for pretty printing.

Usage

```
round_n.ade(x, digits = 0)
```

Arguments

x a numeric R object
digits how many digits are to be shown after decimal?

Details

the function print zeros at the end of a number, to show the precision of rounding

Value

An object of similar structure to x containing character representations of the elements of x in a rounded format

See Also

[format_p.ade](#)

Examples

```
round_n.ade(13.1415, 2)
round_n.ade(3, 3)
```

scatter.ade

*Scatterplot***Description**

Draw a scatter or a bubble plot

Usage

```
scatter.ade(x, y=NULL, group=NULL, z=NULL, data=NULL, vnames=NULL,
            main=NULL, xlab=NULL, ylab=NULL, glab=NULL, zlab=NULL,
            legendon="topright", xlim=NULL, ylim=NULL, zlim=NULL,
            lwd=1, cex=1, pch=16, lty=1,
            xticks=NULL, yticks=NULL, zticks=NULL,
            col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL,
            fitline=0, wall=0, v=NULL, h=NULL, diag=FALSE, span=0.75)
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector of x coordinates for the points • a character string with the name of the x variable in the data.frame • a formula $y \sim x$, $y \sim x + \text{group}$ or $y \sim x + z + \text{group}$
y	<ul style="list-style-type: none"> • a numeric vector of y coordinates for the points • a character string with the name of the y variable in the data.frame
group	<ul style="list-style-type: none"> • a factor to group the points • a character string with the name of the group variable in the data.frame
z	<ul style="list-style-type: none"> • a numeric vector for size of the points • a character string with the name of the size variable in the data.frame
data	data.frame if used character string for (x,y,g,z) or formula
vnames	a vector of character strings with the names of groups in the legend
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
glab	a title of the legend
zlab	a title for the z in the second legend
legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft" • "top"

- "topright"
- "right"
- "center"
- "none"

This places the legend on the inside of the plot frame at the given location. To locate 2 legends you can give a vector of 2 keywords.

xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
zlim	the z limits (z1, z2) for the size of points
lwd	the line width
cex	character (or symbol) expansion: a numerical value, dont work if z is given
pch	plotting "character", i.e., symbol to use. This can either be a single character or an integer code for one of a set of graphics symbols. 15, 16, 17 working well with given z.
lty	the line type
xticks	the number of ticks on the x axis or a vector of exact ticks
yticks	the number of ticks on the y axis or a vector of exact ticks
zticks	the number os Symbols in the z legend or a vector of values for the Symbols
col	a vector of colors for the points for each group
tcol	color of the text in whole plot
bgcol	the background color for plot dekokoration
lcol	color for the lines in plot, a vector of colors is possible
alpha	a parameter in [0, 1] for semi-transparency of points
fitline	a number between 0 and 3 to fit: <ul style="list-style-type: none"> • 0. not fit • 1. a lm regression line • 2. a loess local regression line • 3. a pylinomial regression line
wall	a number between 0 and 6 for selection the dekokoration style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).
diag	logical asking whether to plot a diagonal line
span	the span parameter for lowess curve fit (only if fitline=2)

See Also

[curves.ade](#)

Examples

```
x<-rnorm(1000)
y<-rnorm(1000)
z<-rnorm(1000, 3)
g<-round(runif(1000))
# plot vs ID
scatter.ade(x, vnames=c("blue","red"), alpha=0.25, fitline=2, wall=0, lwd=2, col=4)
# Scatter plot
scatter.ade(x, y*x, g, vnames=c("blue","red"), alpha=0.25, wall=2)
# bubble plot
scatter.ade(x, y, g, z, vnames=c("blue","red"), alpha=0.25, zticks=c(1,2,3,4,5), wall=3)
```

skewness.ade

Simple function to calculate skewness

Description

calculate skewness

Usage

```
skewness.ade(x, na.rm=FALSE, w=NULL)
```

Arguments

x	a numeric vector
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
w	weights

Value

a single number of skewness from x

See Also

[kurtosis.ade](#)

Examples

```
x<-rnorm(1000)
skewness.ade(x)
```

tornado.ade	<i>Tornado or population plot</i>
-------------	-----------------------------------

Description

draw a tornado plot, it could be a population pyramid

Usage

```
tornado.ade(x, group=NULL, group2=NULL, data=NULL, vnames=NULL, gnames=NULL,
            gnames2=NULL, breaks=6, density=NULL, angle=NULL,
            xlab=NULL, glab=NULL, main='', legendon='topright', xticks=NULL,
            col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL, alpha=NULL,
            r=0.05, lwd=1, lty=2, wall=0, v=NULL, h=NULL)
```

Arguments

x	<ul style="list-style-type: none"> • a numeric vector • a single factor • a string with the name of the variable in the data.frame • a formula <code>x~group+group2</code> • a table or matrix • a list of tables
group	<ul style="list-style-type: none"> • a factor to separate the plot in two halves • a string with the name of the factor in the data.frame • nothing if x is a formula, table or list
group2	<ul style="list-style-type: none"> • a factor to separate the plot in several groups • a string with the name of the factor in the data.frame • nothing if x is a formula, table or list
data	a data.frame
vnames	a vector of character strings with the names of groups in tornado eye
gnames	a vector of character strings with the names of both groups
gnames2	a vector of character strings with the names of groups in the legend
breaks	a single number giving the number of cells to separate x, works only if x is a numeric vector
density	the vector of density of shading bars in each group
angle	the vector of slopes of shading bars, given as an angle in degrees (counter-clockwise).
xlab	a title for the x axis
glab	a title for the legend
main	an overall title for the plot

legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft" • "top" • "topright" • "right" • "center" <p>This places the legend on the inside of the plot frame at the given location.</p>
xticks	the number of ticks on the x axis
col	colors for each group
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the lines in plot, a vector of colors is possible, only used if h or v is given
alpha	a parameter in [0, 1] for semi-transparency of bars
r	the width of empty edge for the legend if it overlap the bars
lwd	the line width
lty	the line type, a vector of types is possible
wall	a number between 0 and 6 for selection the decoration style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).

Examples

```
tab1<-cbind(rpois(20, 20),rpois(20, 20))
tab2<-cbind(rpois(20, 15),rpois(20, 15))
tab3<-cbind(rpois(20, 10),rpois(20, 10))
tornado.ade(list(tab1, tab2, tab3), gnames=c('Men','Women'), xlab='number')
```

wall.ade

Plot templates (wall)

Description

A function to make look a like templates of plots for different wall parameters.

Usage

```

wall.ade(vnames=NULL, main=NULL, xlab=NULL, ylab=NULL, glab=NULL,
         legendon="topright", xlim=NULL, ylim=NULL, lwd=1, pch=16, lty=1,
         xticks=NULL, yticks=NULL, col=NULL, tcol=NULL, bgcol=NULL, lcol=NULL,
         wall=0, v=NULL, h=NULL)

```

Arguments

vnames	a vector of character strings with labels in the legend
main	an overall title for the plot
xlab	a title for the x axis
ylab	a title for the y axis
glab	a title of the legend
legendon	a single keyword from: <ul style="list-style-type: none"> • "bottomright" • "bottom" • "bottomleft" • "left" • "topleft" • "top" • "topright" • "right" • "center"

This places the legend on the inside of the plot frame at the given location. To locate 2 legends you can give a vector of 2 keywords.

xlim	the x limits (x1, x2) of the plot
ylim	the y limits (y1, y2) of the plot
lwd	the line width
pch	character or symbol in the legend
lty	the line type
xticks	the number of ticks on the x axis or a vector of exact ticks
yticks	the number of ticks on the y axis or a vector of exact ticks
col	a vector of colors for the points in the legend
tcol	color of the text in whole plot
bgcol	the background color for plot decoration
lcol	color for the lines in plot, a vector of colors is possible
wall	a number between 0 and 6 for selection the decoration style of the plot.
v	the x-value(s) for vertical line(s).
h	the y-value(s) for horizontal line(s).

See Also[scatter.ade](#)**Examples**

```
par(ask=TRUE)
wall.ade(vnames=c('blue', 'red'), wall=0, main='Template of wall 0', xlab='x', ylab='y')
wall.ade(vnames=c('blue', 'red'), wall=1, main='Template of wall 1', xlab='x', ylab='y')
wall.ade(vnames=c('blue', 'red'), wall=2, main='Template of wall 2', xlab='x', ylab='y')
wall.ade(vnames=c('blue', 'red'), wall=3, main='Template of wall 3', xlab='x', ylab='y')
wall.ade(vnames=c('blue', 'red'), wall=4, main='Template of wall 4', xlab='x', ylab='y')
wall.ade(vnames=c('blue', 'red'), wall=5, main='Template of wall 5', xlab='x', ylab='y')
wall.ade(vnames=c('blue', 'red'), wall=6, main='Template of wall 6', xlab='x', ylab='y')
```


Index

- * **AUC**
 - roc.plot.ade, 32
 - * **CI**
 - ratio.plot.ade, 30
 - * **Kaplan-Meier**
 - KM.plot.ade, 21
 - * **Q-Q Plot**
 - qq.ade, 29
 - * **ROC**
 - roc.plot.ade, 32
 - * **Sensitivity**
 - performance.plot.ade, 27
 - * **Specificity**
 - performance.plot.ade, 27
 - * **barplot**
 - bar.plot.ade, 3
 - bar.plot.wtd, 5
 - bar3d.ade, 8
 - * **box-percentile**
 - box.plot.ade, 11
 - * **boxplot**
 - box.plot.ade, 11
 - box.plot.wtd, 13
 - * **bubble**
 - scatter.ade, 34
 - * **correlation**
 - correlogram.ade, 15
 - * **curves**
 - curves.ade, 16
 - * **format**
 - format_n.ade, 18
 - format_p.ade, 19
 - * **histogram**
 - histogram.ade, 20
 - * **kurtosis**
 - kurtosis.ade, 23
 - * **lines**
 - curves.ade, 16
 - * **missings**
 - missiogram.ade, 24
 - * **parallel**
 - parallel.ade, 25
 - parallel.set.ade, 26
 - * **plot**
 - wall.ade, 38
 - * **round**
 - format_n.ade, 18
 - format_p.ade, 19
 - round_n.ade, 33
 - * **scatterplot**
 - scatter.ade, 34
 - * **scatter**
 - bland.altman.ade, 9
 - * **skewness**
 - skewness.ade, 36
 - * **tornado**
 - tornado.ade, 37
 - * **violinlot**
 - box.plot.ade, 11
 - * **wall**
 - wall.ade, 38
-
- bar.plot.ade, 3, 9
 - bar.plot.wtd, 5
 - bar3d.ade, 5, 8, 8
 - bland.altman.ade, 9
 - box.plot.ade, 11
 - box.plot.wtd, 13
 - correlogram.ade, 15
 - curves.ade, 16, 35
 - epade (epade-package), 2
 - epade-package, 2
 - format_n.ade, 18, 19
 - format_p.ade, 18, 19, 33
 - histogram.ade, 20

KM.plot.ade, [21](#)
kurtosis.ade, [23](#), [36](#)

missiogram.ade, [24](#)

parallel.ade, [25](#), [27](#)
parallel.set.ade, [26](#), [26](#)
performance.plot.ade, [27](#)

qq.ade, [29](#)

ratio.plot.ade, [30](#)
roc.plot.ade, [32](#)
round_n.ade, [33](#)

scatter.ade, [11](#), [18](#), [30](#), [34](#), [40](#)
skewness.ade, [23](#), [36](#)

tornado.ade, [37](#)

wall.ade, [38](#)