

Package ‘NicePlots’

August 21, 2018

Type Package

Title Nice Plots for Data Exploration

Version 0.1.1

Author Zachary Hunter

Maintainer Zachary Hunter <zhunter7@gmail.com>

Description More about what it does (maybe more than one line)

Use four spaces when indenting paragraphs within the Description.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends dplyr, beeswarm, tidyr, vioplot

R topics documented:

addNicePoints	2
calcStats	3
drawBar	4
drawBoxPlot	5
drawPoints	6
errorBars	7
facetSpacing	8
formatPlotColors	9
makeColorMatrix	10
makeLogTicks	11
niceBar	12
niceBox	14
niceDots	17
niceVio	20
prepCategoryWindow	22
prepNiceData	24
quantileTrim	25
setAlpha	26

Index	28
--------------	-----------

addNicePoints	<i>Add a datapoint overlay to a box or violin plot</i>
---------------	--

Description

This function prepares data based on settings from [niceBox](#), [niceDots](#), or [niceVio](#) and passes the data on to [drawPoints](#).

Usage

```
addNicePoints(preppedData, by, filter = TRUE, sidePlot = F, subGroup = F,
  plotAt, pointHighlights = F, pointMethod = "jitter", pointShape = 16,
  pointSize = 1, width = 1, pointLaneWidth = 0.9,
  plotColors = formatPlotColors(list(1)), drawPoints = T, outliers = F,
  dataCols = 1)
```

Arguments

preppedData	list; a list object returned by prepCategoryWindow
by	factor or dataframe of factors; One or more factors that control how the data is grouped. The first column is the primary grouping factor and the second and third columns are used for sub-grouping and highlighting as needed.
filter	logical vector; Used to further filter the data if necessary.
sidePlot	logical; switches the axis to plot horizontally instead of vertically.
subGroup	logical; Should the data be faceted into subgroups within the primary factor levels. Ignored if by is a factor .
plotAt	numeric; A vector of where to draw each set of points
pointHighlights	logical; will use additional factors in by to highlight points in the dot plot.
pointMethod	character; method to be used for plotting dots. Can be set to "jitter", "linear", "beeswarm" or "distribution".
pointShape	positive integer; sets pty for plotting data points. Can be a vector to support additional graphical customization.
pointSize	positive integer; sets the cex multiplier for point size.
width	numeric; A multiplier that controls how wide the plotting elements will be. Setting width=1.1 would result in plot elements being 10% wider.
pointLaneWidth	numeric; This controls how far data point dots can move along the categorical axis when plotting. Used for pointMethod options 'jitter', 'beeswarm', and 'distribution'.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
drawPoints	logical; draws a dot plot overlay of the data for each box. Setting this to false causes just the outlier points to be plotted. Used in niceBox .
outliers	positive numeric; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.

`dataCols` numeric; A number of representing the number of data columns to be plotted. These is a combination of the dimentions of `prepedData` and/or the number of primary and secondary grouping factors. Used to determine the maximum plotting width for the points.

Details

This functon takes in cleaned data from `prepCategoryWindow` and reorganizes to to create a dot plot overlay for a graph. This code is used by both `niceBox` and `niceVio` and has been moved to an independant functon to make the code more compact and easier to maintain. This code is also used to draw the outlier dots in a boxplot by setting `drawPoints = FALSE`.

See Also

`drawPoints`, `niceBox`, `niceVio`, `niceDots`, `beeswarm`, `jitter`, `drawPoints`

Examples

```
#Add a beeswarm plot overlay to a boxplot in the iris dataset:
data(iris)
data<-list(data=iris$Sepal.Length)
boxplot(iris$Sepal.Length~iris$Species)
addNicePoints(data,by=iris$Species,pointMethod="beeswarm",plotAt=1:3)

#Add an outlier point to a boxplot:
boxplot(iris$Sepal.Length~iris$Species, outline=FALSE)
addNicePoints(data,by=iris$Species,pointMethod="linear",plotAt=1:3,
  drawPoints=FALSE,outliers=1.5)
```

`calcStats`

calculate preliminary statistical significance analysis

Description

`calcStats` takes a numeric vector and a factor and runs a preliminary statistical analysis. Output is printed to the screen and the p-value is returned as a character string.

Usage

```
calcStats(x, by, type = c("Wilcox", "Tukey", "T.Test", "ANOVA"))
```

Arguments

`x` numeric; numeric vector of data points to analyze.

`by` factor; factor describing the groups within `x` to test.

`type` character; determines which statistical test should be used. Accepted values are 'wilcox', 't.test', 'ttest', 'anova' and 'tukey'. Values not matching a valid input will produce a warning.

Details

This is designed to be used in conjunction with data visualization plots to help with data exploration and should not be used for a robust statistical analysis. Normal distribution, variance and other data characteristics are not evaluated and there is no guarantee that the underlying test assumptions are met. For two level factors `wilcox.test` or `t.test` is recommended. If the factor has more than two levels then `pairwise.wilcox.test` and `pairwise.t.test` are automatically selected. In this case `anova` and the optional follow-up `TukeyHSD` can also be used. All output is printed to the console and for the two level tests and `anova` the p-value is returned as a text string.

Value

a character string describing the test run and the p-value.

See Also

`wilcox.test`, `pairwise.wilcox.test`, `t.test`, `pairwise.t.test`, `anova`, `TukeyHSD`

Examples

```
data(iris)
pv<-calcStats(iris$Petal.Length,by=iris$Species,type="anova")
boxplot(iris$Petal.Length~iris$Species,main="Petal Length by Species",sub=pv)
```

drawBar

drawBar

Description

Add a barplot with options error pars to the active plotting environment

Usage

```
drawBar(x, plotColors, errorBars = FALSE, errorCap = "ball",
  errorLineType = 1, width = 0.5, sidePlot = FALSE, stacked = FALSE,
  capSize = 2, lineWidth = 1)
```

Arguments

x	dataframe; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
errorBars	Logical; Should error bars be drawn. Defaults to true but is ignored if <code>stack=TRUE</code> .
errorCap	character; Determines the style for the ends of the error bars. Valid options are <code>ball</code> , <code>bar</code> or <code>none</code> .
errorLineType	numeric; Sets line type for drawing the error bars.

width	numeric; cex like scaling factor controlling the width of the bars.
sidePlot	logical; Plots bar high on the x axis if set to TRUE .
stacked	logical; draws a stacked barplot if set to TRUE .
capSize	numeric; cex like scaling value the controls the size of the caps on the error bars.
lineWidth	numeric; Sets the lwd options for controlling line plotting thickness for the bar plot.

Details

This function draws a series of bars based on a dataframe. The expected columns include yt (location top of the bar), yb or bottom of the bar, at indicating where the bar should be drawn, Group which is a unique ID per row, fact which contains an optional stacking factor UpperError for the top of the error bar and LowerError for the location of the bottom of the error bar. The construction of the dataframe is handled automatically from input data by [niceBar](#).

See Also

[barplot](#), [niceBar](#), [errorBars](#)

Examples

```
data(iris)
data<-iris %>% group_by(Species) %>%
  summarize(yt=mean(Sepal.Length), yb=0, UpperError=sd(Sepal.Length),
    LowerError=sd(Sepal.Length)) %>%
  ungroup() %>% select(yt,yb,UpperError,LowerError,Group=Species) %>%
  bind_cols(at=1:3,fact=1:3)
plot(type="n",xlim=c(0,4),ylim=c(0,max(iris$Sepal.Length)),-1,xaxt="n")
drawBar(data,plotColors=list())
```

drawBoxPlot	<i>draw a custom box and whisker plot</i>
-------------	---

Description

takes a date frame with columns labeled 'at', 'q1', 'q3', 'min', 'max', 'median' and 'width' to draw a series of boxplots.

Usage

```
drawBoxPlot(x, col = "black", fill = NULL, drawBox = T, drawDot = F,
  whiskerLty = 2, side = FALSE, lWidth = 1)
```

Arguments

x	named list or data frame; x\$at, x\$q2, x\$q4, x\$median, x\$min, x\$max and x\$width must all be defined as numeric vectors in a named list or data.frame object.
col	character; color vector that controls the line color.
fill	character; color vector that determines the interior color of the box.

drawBox	logical; draws the box and whiskers if set to TRUE . The median line will be drawn regardless.
drawDot	logical; draws a circle at the center of the median bar if set to TRUE .
whiskerLty	positive integer; sets the line type or lty option for plotting the whiskers.
side	logical; if set to TRUE , the box plots will be drawn horizontally.
lwidth	positive integer; corresponds to lwd line width setting in base R.

Details

The input data frame x should include columns labels named 'at','q1',and 'q3', 'median', 'min', 'max' and 'width' in any order. Each row will draw a box and whisker plot. The columns 'q1' and 'q3' refer to the 25% and 75% cumulative distribution values that bound the interquartile range. If side=TRUE then the x and y axes are swapped to support horizontal plotting. The box and whiskers can be suppressed leaving only the median line and the optional center marker if so desired.

See Also

[boxplot](#), [niceBox](#)

Examples

```
library(dplyr)
data(iris)
iData<-iris %>% group_by(Species) %>%
  summarize(median=median(Sepal.Length),min=min(Sepal.Length),max=max(Sepal.Length),
    q1=quantile(Sepal.Length)[2],q3=quantile(Sepal.Length)[4]) %>%
  bind_cols(at=c(1:3),width=c(.2,.3,.4))
plot(1,1,type="n",xlim=c(0,4),ylim=c(0,9))
drawBoxPlot(iData)
```

drawPoints	<i>draw dots for a dot plot</i>
------------	---------------------------------

Description

takes a data frame of locations, values and an optional subgrouping factor and adds the data points to the active plot

Usage

```
drawPoints(x, type = "jitter", col = "black", size = 1, shape = 1,
  highlight = FALSE, width = 0.2, sidePlot = FALSE)
```

Arguments

x	named list or data frame; x\$at, x\$data and x\$pfact (optional) should all be defined. These vectors are used to place the the point on the chart and determine the point level grouping (highlighting)
type	character; determines how the points are arranged. Options are 'jitter', 'linear', 'beeswarm' and 'distribution'.

col	character; vector of color names for plotting points. If length is greater than one it will be used for subgroups or will iterate over the groups.
size	numeric; vector of cex values for point size. If length is greater than one it will be used for subgroups or will iterate over the groups.
shape	numeric; vector determining point shapes (pch). If length is greater than one it will be used for subgroups or will iterate over the groups.
highlight	logical; Should the point highlighting option be turned on (assumes that pfact is defined).
width	numeric; determines how far points can deviate from the center category label for type options other than 'linear'.
sidePlot	logical; plots dots for a horizontal rather than vertical axis.

Details

This function adds data points to a chart. These can be organized exactly as specified (linear), as a jitter cloud (jitter), as a waterfall plot (distribution) or as a swarm (beeswarm). A factor labeled pfact can be included in x and used to highlight individual data points by setting subgroup=TRUE. All graphic customization options can given as vectors and will be iterated over during plotting. Note that the size/cex option can not be used to highlight pfact levels in a beeswarm plot and only the first element of the vector will be used.

See Also

[points](#), [stripchart](#), [beeswarm](#)

Examples

```
library(dplyr)
data(iris)
boxplot(iris$Sepal.Length~iris$Species,ylab="Sepal Length")
iData<-data.frame(at=as.numeric(iris$Species),data=iris$Sepal.Length)
drawPoints(iData,type="jitter",col=c("red","blue","purple"))
```

errorBars

draw custom error bars

Description

Draws error bars with an optional cap at one end

Usage

```
errorBars(x, capType = c("none", "bar", "ball"), capSize = NULL,
  side = FALSE, col = "black", lType = 1, width = 1)
```

Arguments

x	named list or data frame; x\$start, x\$stop and x\$at must all be defined as numeric vectors in a named list or data.frame object. In the case of a data frame, each row returns a single error bar.
capType	character; can be set to 'none', 'bar', 'ball'. If set to 'bar' or ball, a round point or a line segment will be used to cap the end of the error bar.
capSize	numeric; capSize is the distance that the cap extends away from the error bar. Set to <code>NULL</code> to suppress the cap regardless of the capType setting.
side	logical; if set to true, the error bars will be drawn horizontally.
col	color; a vector of line colors.
lType	positive integer; corresponds to lty line type in base R.
width	positive numeric; corresponds to lwd line width setting in base R.\#'

Details

The input data frame x should have columns labels 'at', 'start', and 'stop' with at determining the x-axis location and start and stop indicating the position of the segment on the y-axis. If side=TRUE then the x and y axes are swapped to support horizontal plotting. Each row of the data frame will produce one bar and an optional cap can be drawn at the 'stop' location.

Examples

```
library(dplyr)
data(iris)
iData<-iris %>% group_by(Species) %>%
  summarize(Average=mean(Sepal.Length), SD=sd(Sepal.Length))
barplot(iData$Average, ylim=c(0,10), names=levels(iris$Species), ylab="sepal length")
loc<-c(.7,1.9,3.1)
top<-iData$SD*2+iData$Average
bottom<-iData$SD*-2+iData$Average
errorBars(data.frame(at=loc, start=iData$Average, stop=top), capType="ball", capSize=2)
errorBars(data.frame(at=loc, start=iData$Average, stop=bottom), capType="ball", capSize=2)
```

facetSpacing

Generate plotting locations for subgrouping data

Description

facetSpacing generates a vector for the at= specification in functions for data sub-grouping

Usage

```
facetSpacing(subGroup, labels)
```

Arguments

subGroup	positive integer; number of levels in the subgrouping factor
labels	positive integer; number of levels in the primary factor

Details

facetSpacing takes the number factor levels from the primary and secondary grouping factors to generate a vector of positions for plotting subgrouped data for the nicePlots package. The spacing assumes that each primary factor levels is plot on positive integers 1, 2, 3 etc. For a primary factor at position i with f subgroup levels, the subgrouping comes from generating equally spaced intervals starting at $i - \frac{1}{2} + \frac{1}{f+1}$ and ending at $i + \frac{1}{2} - \frac{1}{f+1}$. Simply put:

$$Spacing = \frac{1}{NSubGroups - 1}$$

Value

a numeric vector of where to plot the subgrouped data. Can be supplied to that at= option in plotting functions

See Also

[prepCategoryWindow](#)

Examples

```
boxplot(CNA$BM~ CNA$Status,border="white")
stripchart(CNA$BM~factor(paste0(CNA$Status,CNA$Sex)),add=T,at=facetSpacing(2,2))
```

formatPlotColors	<i>format a NicePlots color list</i>
------------------	--------------------------------------

Description

To simplify code and user options, any color option not set by the user is added to the list and set to the default value.

Usage

```
formatPlotColors(plotColors)
```

Arguments

plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
------------	--

Details

The options NicePlots colors include bg (background color), guides (guide lines for major tick-marks), minorGuides (guide lines for minor tick-marks) lines (lines for box/bar plots etc.), points (plotting data points), fill (fill for box/bar plots etc.), axis (axis colors), majorTick (major tick-mark color), minorTick (minor tick-mark color), labels (label colors), subGroupLabels (sub-group label colors). Any option not set by the user will be added to the list and set to the default in order to insure compatibility with downstream NicePlot functions.

Value

a formatted NicePlots color list.

Examples

```
myCols<-list(bg="lightgrey",fill=c("red","green","blue"),lines="darkgrey")
myCols<-formatPlotColors(myCols)
print(myCols)
```

makeColorMatrix

Create a matrix of increasingly transparent colors

Description

makeColorMatrix is a convenience function for plotting with transparent colors.

Usage

```
makeColorMatrix()
```

Details

This function take no arguments, but generates rows corresponding to red, blue, green, gray, purple and gold with increasing transparency moving from left to right across the columns.

Value

A 6 x 5 matrix of colors.

See Also

[rainbow](#), [col2rgb](#), [rgb](#).

Examples

```
plot(1,1,col="white",xlim=c(0,10),ylim=c(0,10))
for(n in 1:6){rect(0:4,rep(8-n,5),1:5,rep(9-n,5),col=as.matrix(makeColorMatrix())[n,])}

#An example how it can be used in practice:
myData<-rnorm(600)
fact<-factor(c(rep("a",100),rep("b",100),rep("c",100),rep("d",100),rep("e",100),rep("f",100)))
plot(myData,col=makeColorMatrix()[fact,3])
```

makeLogTicks	<i>format a log scale axis</i>
--------------	--------------------------------

Description

Generates the location and labels for the major tick marks for a given log base transformation along with optional minor tick mark location.

Usage

```
makeLogTicks(dataRange, minorCount = 10, logScale = 2, axisText = c(NULL,
  NULL), expLabels = TRUE)
```

Arguments

dataRange	numeric; a numeric vector with the min and max values for the data set prior to log transformation.
minorCount	positive integer; the number of minor tick marks to be drawn between each major tick.
logScale	numeric; the logarithm base to use for the log scale transformation.
axisText	character; a length two character vector containing text to be prepend or append to the major tick labels, respectively.
expLabels	logical; if set to TRUE , the major labels will written as \log_{base}^x . Otherwise the labels will correspond to the non-transformed values at that point.

Details

Base R does not have great visual queues to indicate when data is being plotted in log scale. This is a simple function takes the min and max of the untransformed data and uses [axisTicks](#) from base R to determine the location of the major tick marks in the new scale. To better indicate that the graph is on a log scale, the major tick-marks are labeled in the untransformed values or expressed in as \log_{Scale}^x when `expLabels=TRUE`. The minor tick marks are drawn equidistant from each other between the major tick marks in the untransformed scale giving them shrinking appearance when rendered in log scale coordinates. This can help helps with the interpretation of data within the log scale and adds another visual indication that the data has been transformed. The value of `minorCount` gives number of minor ticks to be drawn between each pair of major tick-marks. `axisText` allows for symbols or units such as ' It is worth stressing again that the input values to `dataRange` are assumed to be raw values prior to log transformation. If log transformed values are given, the axis will be drawn correctly.

Value

a list with the following elements: major tick marks locations `[[1]]`, major tick labels `[[2]]`, minor tick mark locations `[[3]]`.

See Also

[axisTicks](#), [axis](#), [prepCategoryWindow](#)

Examples

```
plot(1:10,log(1:10,2),yaxt="n",ylab="")
majorTicks<-makeLogTicks(c(0,10),minorCount= 4,logScale=2, axisText=c("", "mg"), explLabels=TRUE)
axis(side=2,lab=majorTicks[[2]],at=majorTicks[[1]],las=2)
axis(side = 2, at = majorTicks[[3]], labels = FALSE, tcl = -0.2)
```

niceBar

draw a bar plot

Description

Aggregates data from a numeric vector or dataframe using up to three factors to draw a barplot with optional error bars.

Usage

```
niceBar(x, by = NULL, groupNames = NULL, aggFun = c("mean", "median",
  "none"), errFun = c("sd", "se", "range"), stack = FALSE, main = NULL,
  sub = NULL, ylab = NULL, minorTick = FALSE, guides = TRUE,
  outliers = FALSE, width = 1, errorMultiple = 2, plotColors = list(bg =
  "open", fill = setAlpha("grey", 0.8)), logScale = FALSE, trim = FALSE,
  axisText = c(NULL, NULL), showCalc = FALSE, calcType = "none",
  ylim = NULL, rotateLabels = FALSE, rotateY = TRUE, add = FALSE,
  minorGuides = NULL, extendTicks = TRUE, subGroup = FALSE,
  subGroupLabels = NULL, explLabels = FALSE, sidePlot = FALSE,
  errorBars = TRUE, errorCap = "ball", errorLineType = 1, capSize = 1.2,
  lineWidth = 1.5, ...)
```

Arguments

x	numeric vector or data frame; The input to prepCategoryWindow can be a numeric vector a data frame of numeric vectors.
by	factor or data frame of factors; used as the primary grouping factor and the factor levels will be used as group names if groupNames is not specified. If by is a data frame and subGroup= TRUE , the second column is assumed to be a secondary grouping factor, breaking out the data into sub-categories within each major group determined by the levels of the first column.
groupNames	character vector; overrides the factor levels of by to label the groups
aggFun	character; Determines how the data is summarized by factor level. Valid options are mean, median or none.
errFun	character; How the data spread is characterized by the error bars. Valid options are sd (standard deviation), se (standard error of the mean) or range.
stack	logical; Should one of the factors in by be used make a stacked bar plot. Note that this sort of analysis is nonsensical for many data sets.
main	character; title for the graph which is supplied to the main argument.
sub	character; subtitle for the graph which is supplied to the sub argument. If NULL and showCalc= TRUE it will be used to display the output form calcStats .
ylab	character; y-axis label.

minorTick	positive integer; number of minor tick-marks to draw between each pair of major ticks-marks.
guides	logical; will draw guidelines at the major tick-marks if set to <code>TRUE</code> . Color of the guidelines is determined by <code>plotColors\$guides</code> .
outliers	positive numeric; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.
width	numeric; cex-like scaling factor controlling the width of the bars.
errorMultiple	numeric; How many standard errors/deviations should be represented by the error bars.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
logScale	positive numeric; the base for the for log scale data transformation calculated as $\log(x+1, \logScale)$.
trim	positive numeric; passed to threshold argument of <code>quantileTrim</code> if any data points are so extreme that they should be removed before plotting and downstream analysis. Set to <code>FALSE</code> to disable.
axisText	character; a length two character vector containing text to be prepended or appended to the major tick labels, respectively.
showCalc	logical; if a p-value can be easily calculated for your data, it will be displayed using the sub annotation setting.
calcType	character; should match one of 'none', 'wilcox', 'Tukey', 't.test', 'anova' which will determine which, if any statistical test should be performed on the data.
yLim	numeric vector; manually set the limits of the plotting area (eg. <code>yLim=c(min,max)</code>). Used to format the y-axis by default but will modify the x-axis if <code>side=TRUE</code> .
rotateLabels	logical; sets <code>las=2</code> for the x-axis category labels. Will affect y-axis if <code>side=TRUE</code> . Note that this may not work well if long names or with subgrouped data.
rotateY	logical; sets <code>las=2</code> for the y-axis major tick-mark labels. Will affect x-axis if <code>side=TRUE</code> .
add	logical; causes plotting to be added to the existing plot rather the start a new one.
minorGuides	logical; draws guidelines at minor tick-marks
extendTicks	logical; extends minor tick-marks past the first and last major tick to the edge of the graph provided there is enough room. Works for both log-scale and regular settings.
subGroup	logical; use additional column in <code>by</code> to group the data within each level of the major factor.
subGroupLabels	character vector; sets the labels used for the subGroup factor. Defaults to the levels of the factor.
expLabels	logical; prints the major tick labels is \logScale^x instead of the raw value
sidePlot	logical; switches the axis to plot horizontally instead of vertically.
errorBars	Logical; Should error bars be drawn. Defaults to true but is ignored if <code>stack=TRUE</code> .
errorCap	character; Determines the style for the ends of the error bars. Valid options are <code>ball</code> , <code>bar</code> or <code>none</code> .

errorLineType	numeric; Sets lty line type for drawing the error bars.
capSize	numeric; Controls the cex like scaling of the ball or width of the cap if they are drawn at the end of the error bars for the bar plot.
lineWidth	numeric; Line width for drawing the bar plot.
...	additional options for S3 method variants.

Details

This bar plot function allows for standard barplot features but with error bars, the ability summarize dataframes into bar plots with median/mean values, sort by bar high for waterfall plots, color bars based on interquartile outlier detection and more. Barplots can be clustered by a secondary factor or if a dataframe is passed to x the input values of multiple measurments (dataframe columns) can be clustered together by the primary factor. As with [niceBox](#), [niceDots](#) and [niceVio](#), by can be a factor or a dataframe factors for forming subgroups.

For most data this would be nonsensical but if you data is say store profits by goods by region one could group by region (first)

See Also

[vioplot](#), [boxplot](#), [niceBox](#), [beeswarm](#), [prepCategoryWindow](#)

Examples

```
data(mtcars)
Groups<-data.frame(Cyl=factor(mtcars$cyl),Gear=factor(mtcars$gear))
niceBar(mtcars$mpg,by=Groups,subGroup=TRUE,yLim=c(0,45),main="Mpg by Cylinders and Gear")
```

niceBox

draw a box plot

Description

draws a box plot with optional scatter plot overlays, subgrouping options and log scale support.

Usage

```
niceBox(x, by = NULL, groupNames = NULL, main = NULL, sub = NULL,
  ylab = NULL, minorTick = FALSE, guides = TRUE, outliers = 1.5,
  pointSize = 1, width = 1, pointShape = 16, plotColors = list(bg =
    "open"), logScale = FALSE, trim = FALSE, pointMethod = "jitter",
  axisText = c(NULL, NULL), showCalc = FALSE, calcType = "none",
  drawBox = TRUE, yLim = NULL, rotateLabels = FALSE, rotateY = FALSE,
  add = FALSE, minorGuides = NULL, extendTicks = TRUE, subGroup = FALSE,
  subGroupLabels = NULL, expLabels = TRUE, sidePlot = FALSE,
  drawPoints = TRUE, pointHighlights = FALSE, drawCenterDot = !drawPoints,
  pointLaneWidth = 0.7, flipFacts = FALSE, ...)
```

Arguments

x	numeric vector or data frame; The input to prepCategoryWindow can be a numeric vector a data frame of numeric vectors.
by	factor or data frame of factors; used as the primary grouping factor and the factor levels will be used as group names if groupNames is not specified. If by is a data frame and subGroup=TRUE, the second column is assumed to be a secondary grouping factor, breaking out the data into sub-categories within each major group determined by the levels of the first column.
groupNames	character vector; overrides the factor levels of by to label the groups
main	character; title for the graph which is supplied to the main argument.
sub	character; subtitle for the graph which is supplied to the sub argument. If NULL and showCalc=TRUE it will be used to display the output form calcStats.
ylab	character; y-axis label.
minorTick	positive integer; number of minor tick-marks to draw between each pair of major ticks-marks.
guides	logical; will draw guidelines at the major tick-marks if set to TRUE. Color of the guidelines is determined by plotColors\$guides.
outliers	positive numeric; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.
pointSize	positive integer; sets the cex multiplier for point size.
width	numeric; scaling factor controlling the width of the boxes.
pointShape	positive integer; sets pty for plotting data points. Can be a vector to support additional graphical customization.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
logScale	positive numeric; the base for the for log scale data transformation calculated as $\log(x+1, \logScale)$.
trim	positive numeric; passed to threshold argument of quantileTrim if any data points are so extreme that they should be removed before plotting and downstream analysis. Set to FALSE to disable.
pointMethod	character; method to be used for plotting dots. Can be set to "jitter", "linear", "beeswarm" or "distribution".
axisText	character; a length two character vector containing text to be prepended or appended to the major tick labels, respectively.
showCalc	logical; if a p-value can be easily calculated for your data, it will be displayed using the sub annotation setting.
calcType	character; should match one of 'none', 'wilcox', 'Tukey', 't.test', 'anova' which will determine which, if any statistical test should be performed on the data.
drawBox	logical; should the boxes be drawn. The median bar will be drawn regardless.
yLim	numeric vector; manually set the limits of the plotting area (eg. yLim=c(min,max)). Used to format the y-axis by default but will modify the x-axis if side=TRUE.
rotateLabels	logical; sets las=2 for the x-axis category labels. Will affect y-axis if side=TRUE. Note that this may not work well if long names or with subgrouped data.

rotateY	logical; sets las=2 for the y-axis major tick-mark labels. Will affect x-axis if side=TRUE.
add	logical; causes plotting to be added to the existing plot rather than start a new one.
minorGuides	logical; draws guidelines at minor tick-marks
extendTicks	logical; extends minor tick-marks past the first and last major tick to the edge of the graph provided there is enough room. Works for both log-scale and regular settings.
subGroup	logical; use additional column in by to group the data within each level of the major factor.
subGroupLabels	character vector; sets the labels used for the subGroup factor. Defaults to the levels of the factor.
expLabels	logical; prints the major tick labels as $\log Scale^x$ instead of the raw value
sidePlot	logical; switches the axis to plot horizontally instead of vertically.
drawPoints	logical; draws a dot plot overlay of the data for each box.
pointHighlights	logical; will use additional factors in by to highlight points in the dot plot
drawCenterDot	logical; draws a circle at the middle of the median bar. Will be turned off by default if drawPoints is set to TRUE.
pointLaneWidth	numeric; This controls how far data point dots can move along the categorical axis when plotting. Used for pointMethod options 'jitter', 'beeswarm', and 'distribution'.
flipFacts	logical; When a dataframe of values is given, column names are used as a secondary grouping factor by default. Setting flipFacts=TRUE makes the column names the primary factor and by the secondary factor.
...	additional options for S3 method variants

Details

This box plot function offers extensive log scale support, outlier detection, data point overlay options, data subsetting with a secondary factor, and data point highlighting with a tertiary factor. The complicated part of using this function is handling its many options. A wrapper function to set up and run it with preset options may be a good idea if you are using it along. The function `niceDots` is an example of this. Briefly put, the `by` argument can be a data frame of factors and the function will work through the columns in order as needed. If `x` is a numeric vector, then `by` should be a factor to group it into categories. If `by` is a data frame of factors and `subGroup=TRUE`, then the first column for `by` is used as the grouping factor and the second column is used as the sub-grouping factor. If `pointHighlights==TRUE`, and `subGroup=TRUE`, the third column of `by` is used to highlight points data point overlay (assuming `drawPoints=TRUE`). If `subGroup=FALSE` and `subGroup=TRUE`, then the second column of `by` is used to control the point highlighting. If `x` itself is a data frame of numeric vectors, `subGroup` is automatically set to false and each column of `x` is plotted like a sub-group and grouped by the first column of `by`. Data point highlighting with `pointHighlights=TRUE` can still be used when `x` is a data frame and the highlighting factor will be drawn from the second column of `by`. Please note that the p-values can not always be calculated and are for general exploratory use only. More careful analysis is necessary to determine statistical significance. This function is an S3 generic and can be extended to provide class specific functionality. To further facilitate data exploration, outputs from statistical testing and data set summaries are printed to the console.

See Also

[boxplot](#), [beeswarm](#), [quantileTrim](#), [prepCategoryWindow](#)

Examples

```
data(iris)
mCols<-makeColorMatrix()
myCols<-list(fill=c(mCols[1,3],mCols[2,3],mCols[3,3]),lines="darkblue")
Lab<-"Sepal Length"
niceBox(iris$Sepal.Length,iris$Species,minorTick=4,showCalc=TRUE,
        calcType="anova",ylab=Lab,main="Sepal Length by Species",plotColors=myCols)

plot(density(iris$Petal.Length))
lengthFact<-factor(iris$Petal.Length>2.82,labels=c("short","long"))

Title<-"Sepal Length by Species and Petal Length"
factorFrame<-data.frame(Species=iris$Species,PetalLength=lengthFact)
niceBox(iris$Sepal.Length, by=factorFrame, minorTick=4,subGroup=TRUE,
        ylab=Lab,main=Title,plotColors=myCols)
```

niceDots

draw a dot plot

Description

draws a categorical dot plot with optional data highlighting and log scale support.

Usage

```
niceDots(x, by = NULL, groupNames = NULL, main = NULL, sub = NULL,
        ylab = NULL, minorTick = FALSE, guides = TRUE, outliers = 1.5,
        pointSize = 1, width = 1, pointShape = 1, plotColors = list(bg =
        "open"), logScale = FALSE, trim = FALSE, pointMethod = "beeswarm",
        axisText = c(NULL, NULL), showCalc = FALSE, calcType = "none",
        ylim = NULL, rotateLabels = FALSE, rotateY = FALSE, add = FALSE,
        minorGuides = NULL, extendTicks = TRUE, subGroup = FALSE,
        subGroupLabels = NULL, expLabels = TRUE, sidePlot = FALSE,
        pointHighlights = FALSE, pointLaneWidth = 1, ...)
```

Arguments

x	numeric vector or data frame; The input to prepCategoryWindow can be a numeric vector a data frame of numeric vectors.
by	factor or data frame of factors; used as the primary grouping factor and the factor levels will be used as group names if <code>groupNames</code> is not specified. If <code>by</code> is a data frame and <code>subGroup=TRUE</code> , the second column is assumed to be a secondary grouping factor, breaking out the data into sub-categories within each major group determined by the levels of the first column.
groupNames	character vector; overrides the factor levels of <code>by</code> to label the groups

main	character; title for the graph which is supplied to the main argument.
sub	character; subtitle for the graph which is supplied to the sub argument. If <code>NULL</code> and <code>showCalc=TRUE</code> it will be used to display the output form <code>calcStats</code> .
ylab	character; y-axis label.
minorTick	positive integer; number of minor tick-marks to draw between each pair of major ticks-marks.
guides	logical; will draw guidelines at the major tick-marks if set to <code>TRUE</code> . Color of the guidelines is determined by <code>plotColors\$guides</code> .
outliers	positive numeric; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.
pointSize	positive integer; sets the cex multiplier for point size.
width	numeric; scaling factor controlling the width of the boxes.
pointShape	positive integer; sets pty for plotting data points. Can be a vector to support additional graphical customization.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
logScale	positive numeric; the base for the for log scale data transformation calculated as $\log(x+1, \logScale)$.
trim	positive numeric; passed to threshold argument of <code>quantileTrim</code> if any data points are so extreme that they should be removed before plotting and downstream analysis. Set to <code>FALSE</code> to disable.
pointMethod	character; method to be used for plotting dots. Can be set to "jitter", "linear", "beeswarm" or "distribution".
axisText	character; a length two character vector containing text to be prepended or appended to the major tick labels, respectively.
showCalc	logical; if a p-value can be easily calculated for your data, it will be displayed using the sub annotation setting.
calcType	character; should match one of 'none', 'wilcox', 'Tukey', 't.test', 'anova' which will determine which, if any statistical test should be performed on the data.
ylim	numeric vector; manually set the limits of the plotting area (eg. <code>ylim=c(min,max)</code>). Used to format the y-axis by default but will modify the x-axis if <code>side=TRUE</code> .
rotateLabels	logical; sets <code>las=2</code> for the x-axis category labels. Will affect y-axis if <code>side=TRUE</code> . Note that this may not work well if long names or with subgrouped data.
rotateY	logical; sets <code>las=2</code> for the y-axis major tick-mark labels. Will affect x-axis if <code>side=TRUE</code> .
add	logical; causes plotting to be added to the existing plot rather than start a new one.
minorGuides	logical; draws guidelines at minor tick-marks
extendTicks	logical; extends minor tick-marks past the first and last major tick to the edge of the graph provided there is enough room. Works for both log-scale and regular settings.
subGroup	logical; use additional column in by to group the data within each level of the major factor.

subGroupLabels	character vector; sets the labels used for the subGroup factor. Defaults to the levels of the factor.
explLabels	logical; prints the major tick labels is $\log Scale^x$ instead of the raw value
sidePlot	logical; switches the axis to plot horizontally instead of vertically.
pointHighlights	logical; will use additional factors in by to highlight points in the dot plot
pointLaneWidth	numeric; This controls how far data point dots can move along the categorical axis when plotting. Used for pointMethod options 'jitter', 'beeswarm', and 'distribution'.
...	additional options for S3 method variants

Details

This is a wrapper function for [niceBox](#) that just plots the points with no box distribution data. data point overlay options, data subsetting with a secondary factor, and data point highlighting with a tertiary factor. The complicated part of using this function is handling its many options. A wrapper function to set up and run it with preset options may be a good idea if you are using it along. The function [niceDots](#) is an example of this. Briefly put, the by argument can be a data frame of factors and the function will work through the columns in order as needed. If x is a numeric vector, then by should be a factor to group it into categories. If by is a data frame of factors and subGroup=[TRUE](#), then the first column for by is used as the grouping factor and the second column is used as the sub-grouping factor. If pointHighlights=[TRUE](#), and subGroup=[TRUE](#), the the third column of by is used to highlight points data point overlay (assuming drawPoints=[TRUE](#)). If subGroup=[FALSE](#) and subGroup=[TRUE](#), then the second column of by is used to control the point highlighting. If x itself is a data frame of numeric vectors, subGroup is automatically set to false and each column of x is plotted like a sub-group and grouped by the first column of by. Data point highlighting with pointHighlights=[TRUE](#) can still be used when x is a data frame and the highlighting factor will be drawn from the second column of by. Please note that the p-values can not always be calculated and are for general exploratory use only. More careful analysis is necessary to determine statistical significance. This function is as S3 generic and can be extended to provide class specific functionality. To further facilitate data exploration, outputs from statistical testing and data set summaries are printed to the console.

See Also

[stripchart](#), [beeswarm](#), [quantileTrim](#), [prepCategoryWindow](#), [niceBox](#)

Examples

```
data(iris)
mCols<-makeColorMatrix()
myCols<-list(fill=mCols[1:3,3],lines="darkblue")
niceDots(iris$Sepal.Length,iris$Species,minorTick=4,showCalc=TRUE,calcType="anova",
  ylab="Sepal Length",main="Sepal Length by Species",plotColors=myCols)
```

niceVio	<i>draw a violin plot</i>
---------	---------------------------

Description

draws a violin plot with optional scatter plot overlays, subgrouping options and log scale support.

Usage

```
niceVio(x, h = NULL, medianMarkerShape = 16, by = NULL,
  groupNames = NULL, main = NULL, sub = NULL, ylab = NULL,
  minorTick = FALSE, guides = TRUE, outliers = 1.5, pointSize = 1,
  width = 1, pointShape = 16, plotColors = list(bg = "open"),
  logScale = FALSE, trim = FALSE, pointMethod = "jitter",
  axisText = c(NULL, NULL), showCalc = FALSE, calcType = "none",
  drawBox = TRUE, yLim = NULL, rotateLabels = FALSE, rotateY = FALSE,
  add = FALSE, minorGuides = NULL, extendTicks = TRUE, subGroup = FALSE,
  subGroupLabels = NULL, expLabels = TRUE, sidePlot = FALSE,
  drawPoints = TRUE, pointHighlights = FALSE, pointLaneWidth = 0.7,
  flipFacts = FALSE, ...)
```

Arguments

x	numeric vector or data frame; The input to prepCategoryWindow can be a numeric vector a data frame of numeric vectors.
h	numeric; Used to override the h hight of density estimator setting in vioplot . Default value is NULL .
medianMarkerShape	positive integer; sets the shape type (pch) for the median marker.
by	factor or data frame of factors; used as the primary grouping factor and the factor levels will be used as group names if groupNames is not specified. If by is a data frame and subGroup= TRUE , the second column is assumed to be a secondary grouping factor, breaking out the data into sub-categories within each major group determined by the levels of the first column.
groupNames	character vector; overrides the factor levels of by to label the groups
main	character; title for the graph which is supplied to the main argument.
sub	character; subtitle for the graph which is supplied to the sub argument. If NULL and showCalc= TRUE it will be used to display the output form calcStats .
ylab	character; y-axis label.
minorTick	positive integer; number of minor tick-marks to draw between each pair of major ticks-marks.
guides	logical; will draw guidelines at the major tick-marks if set to TRUE . Color of the guidelines is determined by plotColors\$guides.
outliers	positive numeric; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.
pointSize	positive integer; sets the cex multiplier for point size.

width	numeric; scaling factor controlling the width of the violins.
pointShape	positive integer; sets pty for plotting data points. Can be a vector to support additional graphical customization.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
logScale	positive numeric; the base for the for log scale data transformation calculated as $\log(x+1, \logScale)$.
trim	positive numeric; passed to threshold argument of quantileTrim if any data points are so extreme that they should be removed before plotting and downstream analysis. Set to FALSE to disable.
pointMethod	character; method to be used for plotting dots. Can be set to "jitter", "linear", "beeswarm" or "distribution".
axisText	character; a length two character vector containing text to be prepended or appended to the major tick labels, respectively.
showCalc	logical; if a p-value can be easily calculated for your data, it will be displayed using the sub annotation setting.
calcType	character; should match one of 'none', 'wilcox', 'Tukey', 't.test', 'anova' which will determine which, if any statistical test should be performed on the data.
drawBox	logical; should the interquartile boxes be drawn.
ylim	numeric vector; manually set the limits of the plotting area (eg. <code>ylim=c(min,max)</code>). Used to format the y-axis by default but will modify the x-axis if <code>side=TRUE</code> .
rotateLabels	logical; sets <code>las=2</code> for the x-axis category labels. Will affect y-axis if <code>side=TRUE</code> . Note that this may not work well if long names or with subgrouped data.
rotateY	logical; sets <code>las=2</code> for the y-axis major tick-mark labels. Will affect x-axis if <code>side=TRUE</code> .
add	logical; causes plotting to be added to the existing plot rather than start a new one.
minorGuides	logical; draws guidelines at minor tick-marks
extendTicks	logical; extends minor tick-marks past the first and last major tick to the edge of the graph provided there is enough room. Works for both log-scale and regular settings.
subGroup	logical; use additional column in <code>by</code> to group the data within each level of the major factor.
subGroupLabels	character vector; sets the labels used for the subGroup factor. Defaults to the levels of the factor.
expLabels	logical; prints the major tick labels as \logScale^x instead of the raw value
sidePlot	logical; switches the axis to plot horizontally instead of vertically.
drawPoints	logical; draws a dot plot overlay of the data for each box
pointHighlights	logical; will use additional factors in <code>by</code> to highlight points in the dot plot
pointLaneWidth	numeric; This controls how far data point dots can move along the categorical axis when plotting. Used for pointMethod options 'jitter', 'beeswarm', and 'distribution'.
flipFacts	logical; When a dataframe of values is given, column names are used as a secondary grouping factor by default. Setting <code>flipFacts=TRUE</code> makes the column names the primary factor and <code>by</code> the secondary factor.
...	additional options for S3 method variants

Details

This violin plot function offers extensive log scale support, outlier detection, data point overlay options, data subsetting with a secondary factor, and data point highlighting with a tertiary factor. The complicated part of using this function is handling its many options. A wrapper function to set up and run it with preset options may be a good idea if you are using it along. The function [niceDots](#) is an example of this. Briefly put, the `by` argument can be a data frame of factors and the function will work through the columns in order as needed. If `x` is a numeric vector, then `by` should be a factor to group it into categories. If `by` is a data frame of factors and `subGroup=TRUE`, then the first column for `by` is used as the grouping factor and the second column is used as the sub-grouping factor. If `pointHighlights==TRUE`, and `subGroup=TRUE`, the third column of `by` is used to highlight points data point overlay (assuming `drawPoints=TRUE`). If `subGroup=FALSE` and `subGroup=TRUE`, then the second column of `by` is used to control the point highlighting. If `x` itself is a data frame of numeric vectors, `subGroup` is automatically set to false and each column of `x` is plotted like a sub-group and grouped by the first column of `by`. Data point highlighting with `pointHighlights=TRUE` can still be used when `x` is a data frame and the highlighting factor will be drawn from the second column of `by`. Please note that the p-values can not always be calculated and are for general exploratory use only. More careful analysis is necessary to determine statistical significance. This function is an S3 generic and can be extended to provide class specific functionality. To further facilitate data exploration, outputs from statistical testing and data set summaries are printed to the console.

See Also

[vioplot](#), [boxplot](#), [niceBox](#), [beeswarm](#), [prepCategoryWindow](#)

Examples

```
data(iris)
mCols<-makeColorMatrix()
myCols<-list(fill=c(mCols[1,3],mCols[2,3],mCols[3,3]),lines="darkblue")
Lab<-"Sepal Length"
niceVio(iris$Sepal.Length,by=iris$Species,minorTick=4,showCalc=TRUE,
        calcType="anova",ylab=Lab,main="Sepal Length by Species",plotColors=myCols)

plot(density(iris$Petal.Length))
lengthFact<-factor(iris$Petal.Length>2.82,labels=c("short","long"))

Title<-"Sepal Length by Species and Petal Length"
factorFrame<-data.frame(Species=iris$Species,PetalLength=lengthFact)
niceVio(iris$Sepal.Length, by=factorFrame, minorTick=4,subGroup=TRUE,
        ylab=Lab,main=Title,plotColors=myCols)
```

```
prepCategoryWindow
```

prepare a plotting environment for categorical data such as bar plots or box plots

Description

takes untransformed data and draws the x and y axis with support of subgrouping data within factors, log transformation and outlier trimming.

Usage

```
prepCategoryWindow(x, by = NULL, groupNames = levels(by),
  minorTick = FALSE, guides = TRUE, yLim = NULL, rotateLabels = FALSE,
  rotateY = TRUE, plotColors = list(bg = "open", guides = "black", lines =
  "gray22", points = "darkgrey", fill = "white"), trim = FALSE,
  logScale = FALSE, axisText = c(NULL, NULL), minorGuides = FALSE,
  extendTicks = F, subGroup = FALSE, expLabels = TRUE, sidePlot = FALSE,
  subGroupLabels = NULL, strictLimits = F)
```

Arguments

x	numeric vector or data frame; The input to prepCategoryWindow can be a numeric vector a data frame of numeric vectors.
by	factor or data frame of factors; used as the primary grouping factor and the factor levels will be used as group names if groupNames is not specified. If by is a data frame and subGroup=TRUE, the second column is assumed to be a secondary grouping factor, breaking out the data into sub-categories within each major group determined by the levels of the first column.
groupNames	character vector; overrides the factor levels of by to label the groups
minorTick	positive integer; number of minor tick-marks to draw between each pair of major ticks-marks.
guides	logical; will draw guidelines at the major tick-marks if set to TRUE. Color of the guidelines is determined by plotColors\$guides.
yLim	numeric vector; manually set the limits of the plotting area (eg. yLim=c(min,max)). Used to format the y-axis by default but will modify the x-axis if side=TRUE.
rotateLabels	logical; sets las=2 for the x-axis category labels. Will affect y-axis if side=TRUE. Note that this may not work well if long names or with subgrouped data.
rotateY	logical; sets las=2 for the y-axis major tick-mark labels. Will affect x-axis if side=TRUE.
plotColors	list; a named list of vectors of colors that set the color options for all NicePlot functions. Names left unspecified will be added and set to default values automatically.
trim	positive numeric; passed to threshold argument of quantileTrim if any data points are so extreme that they should be removed before plotting and downstream analysis. Set to FALSE to disable.
logScale	positive numeric; the base for the for log scale data transformation calculated as $\log(x+1, \logScale)$.
axisText	character; a length two character vector containing text to be prepended or appended to the major tick labels, respectively.
minorGuides	logical; draws guidelines at minor tick-marks
extendTicks	logical; extends minor tick-marks past the first and last major tick to the edge of the graph provided there is enough room. Works for both log-scale and regular settings.
subGroup	logical; use additional column in by to group the data within each level of the major factor.
expLabels	logical; prints the major tick labels is \logScale^x instead of the raw value
sidePlot	logical; switches the axis to plot horizontally instead of vertically.

subGroupLabels	character vector; sets the labels used for the subGroup factor. Defaults to the levels of the factor.
strictLimits	logical; eliminates padding on the value axis so 0 can be flush with the x-axis. Defaults to FALSE .

Details

This function does all the hard work of setting up the x and y axis for plotting as well as optionally log transforming and/or trimming the data of outliers. In particular, it adds much more robust support for plotting of log transformed data and subgrouping of primary vectors. Other features include the addition of both major and minor guidelines, support for horizontal plotting and improved label formatting options.

Value

formats the plotting area and returns a named list with 'data' and 'labels' corresponding to the trimmed and/or transformed data and the labels for the primary factors, respectively.

See Also

[axisTicks](#), [axis](#), [makeLogTicks](#), [facetSpacing](#)

Examples

```
todo<-1
```

prepNiceData	<i>Prepare and print basic statistics for niceBox and niceVio</i>
--------------	---

Description

Uses filtered data with subgroup and factor information to calculate quartile data for display and plotting.

Usage

```
prepNiceData(prepedData, by, subGroup = FALSE, outliers = TRUE, filter,
  groupNames, plotLoc, width = 1, flipFacts = FALSE)
```

Arguments

prepedData	list; a list object returned by prepCategoryWindow
by	factor or dataframe of factors; One or more factors that control how the data is grouped. The first column is the primary grouping factor and the second and third columns are used for sub-grouping and highlighting as needed.
subGroup	logical; Should the data be faceted into subgroups within the primary factor levels. Ignored if by is a factor .
outliers	positive numeric; number of interquartile ranges (IQR) past the Q1 (25%) and Q3 (75%) cumulative distribution values. Outliers are often defined as $1.5 \times IQR$ and extreme outliers are more than $3 \times IQR$ away from the inner 50% data range.

filter	logical vector; Used to further filter the data if necessary.
groupNames	character; A character vector for the primary group names
plotLoc	numeric vector; A vector indicating where each element should be plotted
width	numeric; A multiplier that controls how wide the plotting elements will be. Setting width=1.1 would result in plot elements being 10% wider.
flipFacts	logical; When a dataframe of values is given, column names are used as a secondary grouping factor by default. Setting flipFacts=TRUE makes the column names the primary factor and by the secondary factor.

Details

To aid in data interpretation and exploration, quartile distribution statistics are calculated for each group and subgroup if specified. For [niceBox](#) this data is also used to plot the data. The data is parsed by checking outlier and subGroup status as well as checking if either prepData or by are a [data.frame](#) or a [vector](#).

See Also

[niceBox](#), [niceVio](#), [niceDots](#)

Examples

```
data(iris)
filter<-rep(TRUE,length(iris$Species))
loc<-seq(1,length(levels(iris$Species)))
data<-list(data=iris[,1:4])
myData<-prepNiceData(data,by=iris$Species,filter=filter,plotLoc=loc,
  groupNames=levels(iris$Species),outliers=FALSE)
print(myData)
```

quantileTrim	<i>Filter data by interquartile range</i>
--------------	---

Description

quantileTrim takes a numeric vector and removes data points that fall more than threshold * the interquartile range outside of the interquartile range. If returnFilter is set to TRUE then the function returns a named list with the trimmed data and a logical vector

Usage

```
quantileTrim(x, threshold = 3, na.rm = FALSE, returnFilter = FALSE)
```

Arguments

x	a numeric vector or a object compatible with the quantile function
threshold	numeric; the number of interquartile ranges out side of the inner 50% range of the data to use as a cutoff from trimming. Typical values include 1.5 for outliers and 3 for extreme outliers.
na.rm	logical; if true will remove all NA values from x before analyzing the data.

returnFilter logical; will cause the function to return a list including with both the trimmed data and a logical vector that can be used to filter objects of the same length as **x**.

Details

The interquartile range (IQR) also known as the H-spread, represents the range encompassing the middle 50. This is used to as a measure of dispersion around the median and more frequently to detect outlier data points. Here data points are filtered if $x < Q_1 - threshold \times IQR$ and $x > Q_3 + threshold \times IQR$ where Q_1 and Q_3 represent the cumulative 25

Value

The trimmed numeric vector or a returnFilter is **TRUE** then a named list labeled data and filter is returned with the trimmed data and the logical filtering vector, respectively.

See Also

[quantile](#).

Examples

```
x<-rnorm(1000)
paste0(mean(x)," (",range(x),")")
x<-quantileTrim(x,threshold=1.5)
paste0(mean(x)," (",range(x),")")

#Example using the filter function:
myData<-c(NA,rnorm(100),NA,NA,rnorm(100),NA,NA,NA,rnorm(300),NA,10000)
myIndex<-1:508
newData<-quantileTrim(myData,na.rm=TRUE,returnFilter=TRUE)
identical(newData$data,myData[newData$filter])
```

setAlpha

add alpha transparency to a named color

Description

Takes a named color such as "red" or "darkgreen" and adds a level of transparency based on the alpha setting.

Usage

```
setAlpha(x, alpha = 0.2)
```

Arguments

x character string; a text string corresponding to an R color
alpha numeric [0-1]; sets the level of transparency.

Details

setAlpha is a convenience function that uses the [col2rgb](#) and [rgb](#) to add transparency to named colors.

Value

An rgb color with transparency alpha.

See Also

[makeColorMatrix](#), [rainbow](#), [col2rgb](#), [rgb](#).

Examples

```
plot(1,1,col="white",xlim=c(0,10),ylim=c(0,10))  
rect(1,1,7,7,col=setAlpha("darkblue"))  
rect(3,3,9,9, col=setAlpha("red"))
```

Index

addNicePoints, [2](#)
anova, [4](#)
axis, [11](#), [24](#)
axisTicks, [11](#), [24](#)

barplot, [5](#)
beeswarm, [3](#), [7](#), [14](#), [17](#), [19](#), [22](#)
boxplot, [6](#), [14](#), [17](#), [22](#)

calcStats, [3](#), [12](#), [15](#), [18](#), [20](#)
col2rgb, [10](#), [26](#), [27](#)

data.frame, [25](#)
drawBar, [4](#)
drawBoxPlot, [5](#)
drawPoints, [2](#), [3](#), [6](#)

errorBars, [5](#), [7](#)

facetSpacing, [8](#), [24](#)
factor, [2](#), [24](#)
FALSE, [3](#), [13](#), [15](#), [16](#), [18](#), [19](#), [21–24](#)
formatPlotColors, [9](#)

jitter, [3](#)

makeColorMatrix, [10](#), [27](#)
makeLogTicks, [11](#), [24](#)

NA, [25](#)
niceBar, [5](#), [12](#)
niceBox, [2](#), [3](#), [6](#), [14](#), [14](#), [19](#), [22](#), [25](#)
niceDots, [2](#), [3](#), [14](#), [16](#), [17](#), [19](#), [22](#), [25](#)
niceVio, [2](#), [3](#), [14](#), [20](#), [25](#)
NULL, [8](#), [12](#), [15](#), [18](#), [20](#)

pairwise.t.test, [4](#)
pairwise.wilcox.test, [4](#)
points, [7](#)
prepCategoryWindow, [2](#), [3](#), [9](#), [11](#), [14](#), [17](#), [19](#),
[22](#), [22](#), [24](#)
prepNiceData, [24](#)

quantile, [25](#), [26](#)
quantileTrim, [13](#), [15](#), [17–19](#), [21](#), [23](#), [25](#)

rainbow, [10](#), [27](#)
rgb, [10](#), [26](#), [27](#)

setAlpha, [26](#)
stripchart, [7](#), [19](#)

t.test, [4](#)
TRUE, [4–7](#), [11–13](#), [15–23](#), [25](#), [26](#)
TukeyHSD, [4](#)

vector, [25](#)
vioplot, [14](#), [20](#), [22](#)

wilcox.test, [4](#)