Package 'basetheme'

April 7, 2023

Title Themes for Base Graphics Plots

Version 0.1.3

Maintainer Karolis Koncevičius <karolis.koncevicius@gmail.com>

Description Functions to create and select graphical themes for the base plotting system. Contains: 1) several custom pre-made themes 2) mechanism for creating new themes by making persistent changes to the graphical parameters of base plots.

Depends R (>= 3.2.2)

License GPL-2

Encoding UTF-8

URL https://github.com/karoliskoncevicius/basetheme

BugReports https://github.com/karoliskoncevicius/basetheme/issues

RoxygenNote 7.2.3

NeedsCompilation no

Author Karolis Koncevičius [aut, cre]

Repository CRAN

Date/Publication 2023-04-07 16:30:05 UTC

R topics documented:

basetheme 2
colshade 3
lab2col 4
num2col 5

Index **7**

basetheme *Theme Control*

Description

Sets and returns base plotting theme parameters.

Usage

```
basetheme(...)
```

Arguments

`...`	- a sequence of `parameter=value` pairs (see Details).

Details

Function dispatches based on the type of first argument:

1. No arguments - returns the list of the current theme settings.
2. NULL - all theme settings are removed.
3. list - assumed that list stores theme settings.
4. character - a theme with that name is used.
5. parameter=value pair - sets the setting for the specified parameter.

Further arguments can be provided as `parameter=value` pairs. See examples.

The list of theme parameters is always returned invisibly, except when function is called with no arguments.

The parameters set by this function will take presedence over `par()` parameters.

Value

a list of all theme settings (invisibly, unless no arguments were provided).

Author(s)

Karolis Koncevičius

Examples

```
# Set theme by parameters
basetheme(pch=19, las=1, cex=2)
plot(1, 1)

# Obtain list of theme parameters
# for the current theme
basetheme()
# for a specific theme
```

```
theme <- basetheme("brutal")
theme

# Set theme by name
basetheme("dark")
plot(1)

# Set theme by list
theme <- basetheme("clean")
theme$rect.col <- "grey90"
basetheme(theme)
pairs(iris[,1:4], col=iris$Species)

# Reset theme
basetheme(NULL)
```

colshade *Add Shade to Color*

Description

Adds a selected amount of shade or tint to a vector of colors.

Usage

```
colshade(cols, frac = 0)
```

Arguments

cols	a vector of colors
frac	a vector of shade fractions (between -1 and 1, defaults to 0)

Details

This function adds shades and tints to provided list of colors. Shade or tint is decided depending on the sign of the `frac` argument: positive values make colors darker and negative values lighten them.

Value

a vector of colors with added shades or tints.

Author(s)

Karolis Koncevičius

`lab2col` *Labels to Colors*

Description

Assigns colors to the provided vector of labels.

Usage

```
lab2col(x, pal, ref = x, NAcol)
```

Arguments

`x`	vector of labels (always transformed to character)
`pal`	colors used to build the palette (defaults to colors set by theme)
`ref`	reference for assigning colors (defaults to elements of `x`)
`NAcol`	color to be used for NA values (defaults to color set by theme)

Details

This function assigns colors to each unique level of label vector `x`. Mainly used for consistently assigning colors to a unique set of labels, especially when the order of original labels might change (see examples).

Color of NA values and values outside of specified reference can be set using `NAcol` argument. Set this to NA to omit the display of such values.

`ref` parameter can be specified to select the order of color assignment: first element from `ref` will be assigned the first color from `pal`, second element - second color, and so on.

The list of provided colors is expanded by first adding shades and then adding tints. However if the number of groups exceeds the number of provided colors by more than 3 times the colors will be repeated.

When `x` is not specified - a function that generates colors based on `pal` and `ref` is returned.

Value

a vector of colors for each element in `x` or, when `x` is missing, a function.

Author(s)

Karolis Koncevičius

See Also

`num2col`

Examples

```
# iris example
pairs(iris[,1:4], col=lab2col(iris$Species))

# iris example with one group missing from reference
pairs(iris[,1:4], col=lab2col(iris$Species, ref=c("setosa", "versicolor")))

# example of using a coloring function
# "color" function below will consistently assign colors to values in "ref".
color <- lab2col(ref=unique(chickwts$feed))
par(mfrow=c(1,2))
means <- tapply(chickwts$weight, chickwts$feed, mean)
barplot(means, col=color(names(means)), las=2)
means <- sample(means)
barplot(means, col=color(names(means)), las=2)
```

---

num2col *Numbers to Colors*

Description

Assigns colors to the provided vector of numbers.

Usage

```
num2col(x, pal, ref = range(x, na.rm = TRUE), NAcol)
```

Arguments

x	numeric vector (factors are transformed to numeric)
pal	colors used to build the palette (defaults to colors set by theme)
ref	reference for assigning colors (defaults to the range of x)
NAcol	color to be used for NA values (defaults to color set by theme)

Details

This function interpolates a given set of colors to a numeric vector. Main use case is in turning numbers into colors for plots, especially when different ranges of x have to be colored differently.

Color of NA values and values outside of ref range can be set using NAcol argument. Set this to NA to omit the display of such values.

In case only a single color is provided - it is expanded by using tints and shades.

When x is not specified - a function that generates colors based on pal and ref is returned.

Value

a vector of colors for each element in x or, when x is missing, a function.

Author(s)

Karolis Koncevičius

See Also

`lab2col`

Examples

```
# color numbers by y-axos
plot(mtcars$hp, mtcars$mpg, col=num2col(mtcars$mpg), pch=19)

# color only a certain range
plot(mtcars$hp, mtcars$mpg, col=num2col(mtcars$mpg, ref=c(20, 35)), pch=19)

# hide the out of range values
plot(mtcars$hp, mtcars$mpg, col=num2col(mtcars$mpg, ref=c(20,35), NAcol=NA))

# iris example
pairs(iris[,-5], col=num2col(iris$Sepal.Length))

# same butusing a prepared coloring function (for values in range 0-10)
color <- num2col(ref=c(0,10))
pairs(iris[,-5], col=color(iris$Sepal.Length))
```

Index

basetheme, 2

colshade, 3

lab2col, 4

num2col, 5