Package: colortools (via r-universe)

November 16, 2026

Type Package

Title Tools for Colors in a Hue-Saturation-Value (HSV) Color Model

Version 0.1.5.9001

Author Gaston Sanchez [aut, cre]

Maintainer Gaston Sanchez <gaston.stat@gmail.com>

Description R package with handy functions to help users select and play with color schemes in an HSV color model. Compatibility updates by Renova.

License GPL-3

URL http://www.gastonsanchez.com

Collate 'adjacent.R' 'anagrams.R' 'analogous.R' 'col2HSV.R' 'complementary.R' 'opposite.R' 'pals.R' 'pizza.R' 'sequential.R' 'setColors.R' 'splitComp.R' 'square.R' 'tetradic.R' 'triadic.R' 'wheel.R'

NeedsCompilation no

Repository https://renova-r.r-universe.dev

Date/Publication 2026-11-16 17:54:27 UTC

RemoteUrl https://github.com/renova-r/colortools

RemoteRef renova-0.1.5.9001

RemoteSha 1cb82798c399ea412be72bb6c47a3b05832832db

Contents

adjacent 2
anagrams 3
col2HSV 4
complementary 5
pals 6
pizza 7
sequential 8
setColors 9

splitComp 10
square 11
tetradic 12
triadic 13
wheel 14

Index **16**

`adjacent` *Adjacent or analogous colors*

Description

Adjacent color schemes use colors that are next to each other on the color wheel. These colors usually match well and create comfortable designs.

Usage

```
adjacent(color, plot = TRUE, bg = "white", labcol = NULL,
  cex = 0.8, title = TRUE)
```

Arguments

`color`	an R color name or a color in hexadecimal notation
`plot`	logical value indicating whether to plot a color wheel with the generated scheme
`bg`	background color of the plot. Used only when `plot=TRUE`
`labcol`	color for the labels (i.e. names of the colors). Used only when `plot=TRUE`
`cex`	numeric value indicating the character expansion of the labels
`title`	logical value indicating whether to display a title in the plot. Used only when `plot=TRUE`

Details

The analogous colors are obtained following a color wheel with 12 colors, each one spaced at 30 degrees from each other.

Value

A character vector with the given color and the analogous colors in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

`complementary`, `splitComp`, `triadic`, `tetradic`, `square`

Examples

```
# analogous colors of 'red'
adjacent("red", plot = FALSE)

# analogous colors of 'tomato' with default color wheel
analogous("tomato")

# analogous colors of '#606FEF' with darker background
adjacent("#606FEF", bg = "gray20")
```

`anagrams` *anagram colors*

Description

This function returns the anagrams of a given color. Anagrams are made by rearranging the hexadecimal letter pairs to form another color.

Usage

```
anagrams(color)
```

Arguments

`color`	an R color name or a color in hexadecimal notation

Details

An anagram is a word that is made by rearranging the letters of another word. In this case, we have color names instead of words, and hexadecimal letter-pairs instead of single letters. For example, the color "#CD1091" is composed by three pairs "CD", "10", and "91". By rearranging these three pairs, we could obtain an anagram like "#CD9110" or "#10CD91"

Value

A character vector with the anagram colors of a given color in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

`pizza`

Examples

```
# anagrams of 'tomato'
ana_tom <- anagrams("tomato")

# plot tomato anagrams in a color 'pizza' wheel
pizza(ana_tom, bg = "gray20")
```

---

col2HSV *col2HSV: converts a color to HSV in hexadecimal notation*

Description

col2HSV converts an R color (or a set of colors) into an HSV color model, and then returns the color names in hexadeciaml notation

Usage

```
col2HSV(color)
```

Arguments

color	an R color name or a color in hexadecimal notation

Value

A character vector with the color(s) name(s) in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

wheel

Examples

```
# convert 'tomato'
col2HSV("tomato")
```

`complementary` *Complementary or opposite color*

Description

Complementary or opposite color scheme is formed by colors that are opposite each other on the color wheel (example: red and green). The high contrast of complementary colors creates a vibrant look that must be managed well so it is not jarring.

Usage

```
complementary(color, plot = TRUE, bg = "white",
  labcol = NULL, cex = 0.8, title = TRUE)
```

Arguments

`color`	an R color name or color in hexadecimal notation
`plot`	logical value indicating whether to plot a color wheel with the generated scheme
`bg`	background color of the plot. Used only when `plot=TRUE`
`labcol`	color for the labels (i.e. names of the colors). Used only when `plot=TRUE`
`cex`	numeric value indicating the character expansion of the labels
`title`	logical value indicating whether to display a title in the plot. Used ony when `plot=TRUE`

Details

The complementary color is obtained following a color wheel with 12 colors, each one spaced at 30 degrees from each other. Complementary color schemes are tricky to use in large doses, but work well when you wnat something to stand out. In addition, omplementary colors are really bad for text.

Value

A character vector with the given color and the complementary color in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

`adjacent`, `splitComp`, `triadic`, `tetradic`, `square`

Examples

```
# complementary color of 'tomato' with no plot
opposite("tomato", plot = FALSE)

# complementary color of 'tomato' with color wheel
opposite("tomato", bg = "gray30")
```

`pals` *Palette colors included in colortools*

Description

This function can be used to display the palettes included in the package, or to show the color names of the specified palette

Usage

```
pals(name = NULL, bg = "white")
```

Arguments

`name`	optional argument to specify the name of the palette
`bg`	background color of the plot. Used only when `name=NULL` (by default)

Details

When no name is given, a plot window will show the different palettes. If a name is given, then the function will return the names of the colors for that palette (no plot)

Author(s)

Gaston Sanchez

See Also

`wheel`

Examples

```
# default use
pals()

# color names of palette 'terra'
pals("terra")

# pizza wheel for palette 'cheer'
pizza(pals("cheer"), init.angle = 90)

# color wheel for first color in palette 'fish'
wheel(pals("fish")[1])
```

pizza *Pizza color wheel*

Description

This function displays a color wheel with specified colors

Usage

```
pizza(colors, bg = "gray95", border = NA,
  init.angle = 105, cex = 0.8, lty = 1, labcol = NULL,
  ...)
```

Arguments

colors	a vector with R color names of colors in hexadecimal notation
bg	background color of the plot. Default "gray95"
border	color of the border separating the pizza slices
init.angle	integer value indicating the start angle (in degrees) for the slices
cex	numeric value indicating the character expansion of the labels
lty	argument passed to polygon which draws each slice
labcol	color for the labels (i.e. names of the colors)
...	graphical parameters (par) can be given as argument to pizza

Details

This function is based on the pie function

Author(s)

Gaston Sanchez

See Also

wheel

Examples

```
# pizza color wheel for rainbow colors
pizza(rainbow(7))

# pizza color wheel for tomato (18 colors)
pizza(setColors("tomato", 18), bg = "gray20", cex = 0.7)
```

`sequential` *sequential HSV colors*

Description

This functions allows to get a sequence of colors in an HSV model with optional pre-especified numbers for saturation, value, and alpha. It is a very flexible function to play with different combinations of saturation, value, and alpha.

Usage

```
sequential(color, percentage = 5, what = "saturation",
  s = NULL, v = NULL, alpha = NULL, fun = "linear",
  plot = TRUE, verbose = TRUE)
```

Arguments

`color`	an R color name or a color in hexadeciaml notation
`percentage`	numeric value indicating the increment steps of the sequence in percentage
`what`	character string indicating what parameter to taki into account to generate the sequence. Possible values are `"saturation"`, `"value"`, and `alpha`
`s`	optional decimal value (between 0 and 1) to fix the color saturation
`v`	optional decimal value (between 0 and 1) to fix the color value
`alpha`	optional decimal value (between 0 and 1) to fix the color alpha transparency
`fun`	character string indicating the applied transformation to the generated sequence. Possible values are `"linear"`, `"sqrt"`, and `"log"`
`plot`	logical value indicating whether to plot the sequence
`verbose`	logical value indicating whether to return the color names of the sequence

Details

The idea bechind this function is to explore a sequence of colors given some fixed numbers of saturation, valur or alpha for an HSV color model. The argument `what` will be taken to generate the sequence in the given `percentage` increment steps. In addition, we can specify a number for `s, v, alpha`. For example, if `what="value"`, we can fix the saturation in `s=0.8`, obtaining a sequence of colors with different values but with the same level of saturation.

The argument `fun` allows to apply a transformation to the generated sequence. By default `fun="linear"`, no transformation is applied. If `fun="sqrt"`, the square root of the generated sequence will be taken. If `fun="log"`, the logarithmic of the generated sequence will be taken.

Author(s)

Gaston Sanchez

See Also

pizza

Examples

```
# sequence for 'orange'
sequential("orange")

# sequence for 'orange' with fun='sqrt' transformation
sequential("orange", fun = "sqrt")

# sequence for 'orange' with fun='log' transformation
sequential("orange", fun = "log")

# sequential sequence for value with fix saturation s=0.7 and fun='log'
sequential("orange", what = "value", s = 0.7, fun = "log")

# sequential sequence for saturation, with fix value s=0.8, alpha=0.5, percentage 10, and fun='log'
sequential("orange", 10, what = "value", s = 0.7, alpha = 0.5, fun = "log")
```

---

setColors *Set Colors for a color wheel*

Description

This function set a given number of colors to create a color wheel

Usage

```
setColors(color, num)
```

Arguments

color	an R color name or a color in hexadecimal notation
num	integer value indicating how many colors to be added to the wheel

Value

A character vector with the given color and the set of colors to create a wheel color

Author(s)

Gaston Sanchez

See Also

col2HSV

Examples

```
# create a color wheel based on 'tomato'
setColors("tomato", 12)

# set 7 colors for '#3D6DCC'
setColors("#3D6DCC", 7)
```

---

splitComp *Split Complementary Color Scheme*

Description

The split-complementary color scheme is a variation of the complementary color scheme. It uses the two colors adjacent to its complement.

Usage

```
splitComp(color, plot = TRUE, bg = "white",
  labcol = NULL, cex = 0.8, title = TRUE)
```

Arguments

color	an R color name of a color in hexadecimal notation
plot	logical value indicating whether to plot a color wheel with the generated scheme
bg	background color of the plot. Used only when plot=TRUE
labcol	color for the labels (i.e. names of the colors). Used only when plot=TRUE
cex	numeric value indicating the character expansion of the labels
title	logical value indicating whether to display a title in the plot. Unsed only when plot=TRUE

Details

This color scheme has the same strong visual contrast as the complementary scheme, but it is supposed to have less tension.

Value

A character vector with the given color and the split-complementary colors in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

complementary

Examples

```
# split-complementary colors of 'tomato' (no plot)
splitComp("tomato", plot = FALSE)

# split-complementary colors of 'tomato' (with dark gray background)
splitComp("tomato", bg = "gray40")
```

`square` *Square color scheme*

Description

The square color scheme is similar to the tetradic scheme, but with all four colors spaced around the color circle.

Usage

```
square(color, plot = TRUE, bg = "white", labcol = NULL,
  cex = 0.8, title = TRUE)
```

Arguments

`color`	an R color name or a color in hexadecimal notation
`plot`	logical value indicating whether to plot a color wheel with the generated scheme
`bg`	background color of the plot. Used only when `plot=TRUE`
`labcol`	color for the labels (i.e. names of the colors). Used only when `plot=TRUE`
`cex`	numeric value indicating the character expansion of the labels
`title`	logical value indicating whether to display a title in the plot. Used only when `plot=TRUE`

Details

The square colors are obtained following a color wheel with 12 colors, each one spaced at 30 degrees from each other. Square color schemes tend to work best if you let one color be dominant

Value

A character vector with the given color and the square scheme colors in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

`complementary`, `adjacent`, `triadic`, `tetradic`

Examples

```
# square color scheme for 'tomato'
square("tomato")
```

`tetradic` *Tetradic Color Scheme*

Description

Tetradic color schemes uses four colors arranged into two complementary pairs.

Usage

```
tetradic(color, plot = TRUE, bg = "white", labcol = NULL,
  cex = 0.8, title = TRUE)
```

Arguments

`color`	an R color name or a color in hexadecimal notation
`plot`	logical value indicating whether to plot a color wheel with the generated scheme
`bg`	background color of the plot. Used only when `plot=TRUE`
`labcol`	color for the labels (i.e. names of the colors). Used only when `plot=TRUE`
`cex`	numeric value indicating the character expansion of the labels
`title`	logical value indicating whether to display a title in the plot. Used only when `plot=TRUE`

Details

The tetradic colors are obtained following a color wheel with 12 colors, each one spaced at 30 degrees from each oter.

Value

A character vector with the given color and the tetradic colors in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

`complementary`, `splitComp`, `adjacent`, `triadic`, `square`

Examples

```
# tetradic colors for 'tomato'
tetradic("tomato")

# tetradic colors for 'tomato' with bg='gray20'
tetradic("tomato", bg = "gray20")
```

`triadic` *Triadic Color Scheme*

Description

Triadic color schemes use colors that are evenly spaced around the color wheel.

Usage

```
triadic(color, plot = TRUE, bg = "white", labcol = NULL,
  cex = 0.8, title = TRUE)
```

Arguments

`color`	an R color name or a color in hexadecimal notation
`plot`	logical value indicating whether to plot a color wheel with the generated scheme
`bg`	background color of the plot. Used only when `plot=TRUE`
`labcol`	color for the labels (i.e. names of the colors). Used only when `plot=TRUE`
`cex`	numeric value indicating the character expansion of the labels
`title`	logical value indicating whether to display a title in the plot. Used only when `plot=TRUE`

Details

The triadic colors are obtained following a color wheel with 12 colors, each one spaced at 30 degrees from each other. Triadic color schemes tend to be quite vibrant. To use a triadic harmony successfully, the colors should be carefully balanced letting one color dominate and use the others for accent.

Value

A character vector with the given color and the triadic colors in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

`complementary`, `splitComp`, `adjacent`, `tetradic`, `square`

Examples

```
# triadic colors of 'tomato'
triadic("tomato")

# triadic colors of 'tomato' with background color 'gray20'
triadic("tomato", bg = "gray20")
```

`wheel` *Color Wheel*

Description

This function generates a color wheel for a given color

Usage

```
wheel(color, num = 12, bg = "gray95", border = NULL,
  init.angle = 105, cex = 1, lty = NULL, main = NULL,
  verbose = TRUE, ...)
```

Arguments

`color`	an R color name or a color in hexadecimal notation
`num`	integer value indicating how many colors to be generated for the color wheel
`bg`	background color of the plor
`border`	color of the border separating the slices
`init.angle`	integer value indicating the start angle (in degrees) for the slices
`cex`	numeric value indicating the character expansion of the labels
`lty`	argument passed to `polygon` which draws the slices
`main`	an overall title for the plot
`verbose`	logical value indicating whether to return the color names
`...`	graphical parameters (`par`) can be given as argument to `wheel`

Details

This function is based on the `pie` function

Value

A character vector with the color names of the generated wheel in hexadecimal notation

Author(s)

Gaston Sanchez

See Also

pizza

Examples

```
# wheel color with 18 slices for 'tomato'
wheel("tomato", num = 18, bg = "gray20", cex = 0.7)
```

Index

adjacent, 2, *5*, *11–13*
anagrams, 3
analogous *(*adjacent*)*, 2

col2HSV, 4, *9*
complementary, *2*, 5, *10–13*

opposite *(*complementary*)*, 5

pals, 6
par, *7*, *14*
pie, *7*, *14*
pizza, *3*, 7, *9*, *15*
polygon, *7*, *14*

sequential, 8
setColors, 9
splitComp, *2*, *5*, 10, *12, 13*
square, *2*, *5*, 11, *12, 13*

tetradic, *2*, *5*, *11*, 12, *13*
triadic, *2*, *5*, *11, 12*, 13

wheel, *4*, *6, 7*, 14