Package 'colorRamp2'

December 21, 2022

Type Package

Title Generate Color Mapping Functions

Version 0.1.0

Date 2022-12-21

Description A color mapping is generated according to the break values and corresponding colors. Other colors are generated by interpolating in a certain color space. The functions were part of the 'circlize' package <https://CRAN.R-project.org/package=circlize>.

Depends R (> 3.0.0)

Imports colorspace, grDevices, methods, stats

Suggests knitr

VignetteBuilder knitr

URL https://github.com/jokergoo/colorRamp2

License MIT + file LICENSE

NeedsCompilation no

Author Zuguang Gu [aut, cre] (<https://orcid.org/0000-0002-7395-8709>)

Maintainer Zuguang Gu <z.gu@dkfz.de>

Repository CRAN

Date/Publication 2022-12-21 16:50:02 UTC

R topics documented:

add_transparency 2
col2value 2
colorRamp2 3
rand_color 4

Index **6**

add_transparency *Add transparency to colors*

Description

Add transparency to colors

Usage

```
add_transparency(col, transparency = 0)
```

Arguments

col	A vector of colors.
transparency	Transparency, numeric value between 0 and 1.

Value

A vector of colors.

Examples

```
add_transparency("red", 0.5)
add_transparency(1, 0.5)
add_transparency("#FF000080", 0.2)
```

col2value *Convert back from colors to values*

Description

Convert back from colors to values

Usage

```
col2value(r, g, b, col_fun)
```

Arguments

r	Red channel in sRGB color space. Value should be between 0 and 1. The value can also be a character vector of colors or a three-column matrix with r, g, b as columns. In this case, g and b are ignored,
g	Green channel in sRGB color space. Value should be between 0 and 1.
b	Blue channel in sRGB color space. Value should be between 0 and 1.
col_fun	the color mapping function generated by colorRamp2.

Details

colorRamp2 maps values to colors and this function does the reversed job. Note for some color spaces, it cannot convert back to the original value perfectly.

Value

A vector of original numeric values.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
x = seq(0, 1, length.out = 11)
col_fun = colorRamp2(c(0, 0.5, 1), c("blue", "white", "red"))
col = col_fun(x)
col2value(col, col_fun = col_fun)
col2value("red", col_fun = col_fun)

col_fun = colorRamp2(c(0, 0.5, 1), c("blue", "white", "red"), space = "sRGB")
col = col_fun(x)
col2value(col, col_fun = col_fun)
```

---

colorRamp2 *Generate color mapping functions*

Description

Generate color mapping functions

Usage

```
colorRamp2(breaks, colors, transparency = 0, space = "LAB",
    hcl_palette = NULL, reverse = FALSE)
```

Arguments

breaks	A vector of numeric break values.
colors	A vector of colors which correspond to values in breaks.
transparency	A single value in [0, 1]. 0 refers to no transparency and 1 refers to full transparency.
space	Color space in which colors are interpolated. Value should be one of "RGB", "LAB", "XYZ", "sRGB", "LUV", see color-class for details.
hcl_palette	Name of the HCL palette. Value should be supported in hcl.pals.
reverse	Whether should the colors in hcl_palette be reversed.

Details

Colors are linearly interpolated according to the break values and corresponding colors through a certain color space. Values exceeding breaks will be assigned with corresponding maximum or minimum colors.

Value

A function which accepts a vector of numeric values and returns interpolated colors.

See Also

`col2value` converts back to the original values by providing the color mapping function generated by `colorRamp2`.

Examples

```
col_fun = colorRamp2(c(-1, 0, 1), c("green", "white", "red"))
col_fun(c(-2, -1, -0.5, 0, 0.5, 1, 2))
```

---

rand_color *Generate random colors*

Description

Generate random colors

Usage

```
rand_color(n, hue = NULL, luminosity = "random", transparency = 0, friendly = FALSE)
```

Arguments

`n`	Number of colors
`hue`	The hue of the generated color. You can use following default color name: `red`, `orange`, `yellow`, `green`, `blue`, `purple`, `pink` and `monochrome`. If the value is a hexidecimal color string such as `#00FFFF`, the function will extract its hue value and use that to generate colors.
`luminosity`	it controls the luminosity of the generated color. The value should be a string containing `bright`, `light`, `dark` and `random`.
`transparency`	Transparency, numeric value between 0 and 1.
`friendly`	If it is true, light random colors will not be generated.

Details

The code is adapted from randomColor.js (`https://github.com/davidmerfield/randomColor`).

Value

A vector of colors.

Author(s)

Zuguang Gu <z.gu@dkfz.de>

Examples

```
plot(NULL, xlim = c(1, 10), ylim = c(1, 8), axes = FALSE, ann = FALSE)
points(1:10, rep(1, 10), pch = 16, cex = 5,
    col = rand_color(10))
points(1:10, rep(2, 10), pch = 16, cex = 5,
    col = rand_color(10, luminosity = "bright"))
points(1:10, rep(3, 10), pch = 16, cex = 5,
    col = rand_color(10, luminosity = "light"))
points(1:10, rep(4, 10), pch = 16, cex = 5,
    col = rand_color(10, luminosity = "dark"))
points(1:10, rep(5, 10), pch = 16, cex = 5,
    col = rand_color(10, hue = "red", luminosity = "bright"))
points(1:10, rep(6, 10), pch = 16, cex = 5,
    col = rand_color(10, hue = "green", luminosity = "bright"))
points(1:10, rep(7, 10), pch = 16, cex = 5,
    col = rand_color(10, hue = "blue", luminosity = "bright"))
points(1:10, rep(8, 10), pch = 16, cex = 5,
    col = rand_color(10, hue = "monochrome", luminosity = "bright"))
```

Index

add_transparency, 2

col2value, 2, *4*
colorRamp2, *2*, *3*, 3, *4*

hcl.pals, *3*

rand_color, 4

sRGB, *2*