

# Package ‘colourvalues’

November 7, 2018

**Type** Package

**Title** Assigns Colours to Values

**Version** 0.2.0

**Date** 2018-11-07

**Description** Maps one of the viridis colour palettes, or a user-specified palette to values. Viridis colour maps are created by Stéfan van der Walt and Nathaniel Smith. They were set as the default palette for the 'Python' 'Matplotlib' library, introduced at SciPy 2015 conference <<http://scipy2015.scipy.org/ehome/index.php?eventid=115969&>>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**LinkingTo** Rcpp, BH

**Imports** Rcpp

**RoxygenNote** 6.1.0

**Suggests** covr, microbenchmark, ggplot2, scales, testthat, viridisLite

**NeedsCompilation** yes

**Author** David Cooley [aut, cre]

**Maintainer** David Cooley <dcooley@symbolix.com.au>

**Repository** CRAN

**Date/Publication** 2018-11-07 05:10:03 UTC

## R topics documented:

cividis . . . . .	2
colour_values . . . . .	2
colour_values_rgb . . . . .	5
convert_colour . . . . .	7
inferno . . . . .	7
magma . . . . .	8
plasma . . . . .	8
viridis . . . . .	8

**Index****9**


---

cividis	<i>Cividis</i>
---------	----------------

---

**Description**

Data frame of the cividis palette

**Usage**

```
cividis()
```

---

colour_values	<i>Colour Values</i>
---------------	----------------------

---

**Description**

maps colours to values

**Usage**

```
colour_values(x, palette = "viridis", na_colour = "#808080FF",
  alpha = 255, include_alpha = TRUE, ...)
```

```
color_values(x, palette = "viridis", na_colour = "#808080FF",
  alpha = 255, include_alpha = TRUE, ...)
```

```
## S3 method for class 'character'
colour_values_to_hex(x, palette, na_colour, alpha,
  include_alpha, summary = FALSE)
```

```
## Default S3 method:
colour_values_to_hex(x, palette, na_colour, alpha,
  include_alpha, n_summaries = 0, format = TRUE, digits = 2)
```

```
## S3 method for class 'logical'
colour_values_to_hex(x, palette, na_colour, alpha,
  include_alpha, summary = FALSE)
```

```
## S3 method for class 'factor'
colour_values_to_hex(x, palette, na_colour, alpha,
  include_alpha, summary = FALSE)
```

```
## S3 method for class 'Date'
colour_values_to_hex(x, palette, na_colour, alpha,
```

```

include_alpha, n_summaries = 0, format = TRUE)

## S3 method for class 'POSIXct'
colour_values_to_hex(x, palette, na_colour, alpha,
  include_alpha, n_summaries = 0, format = TRUE)

## S3 method for class 'POSIXlt'
colour_values_to_hex(x, palette, na_colour, alpha,
  include_alpha, n_summaries = 0, format = TRUE)

```

## Arguments

x	vector of values to map to a colour
palette	colour palette. See details and examples
na_colour	hex string colour to use for NA values in the form #RRGGBBAA.
alpha	optional. Single value in [0,255] applied to all colours, or a vector of numeric values the same length as x and at least length 5. The numeric vector will be scaled into the range [0,255]. If a matrix palette is supplied this argument is ignored.
include_alpha	logical indicating if the returned hex or matrix should include the alpha values. Defaults to TRUE.
...	other arguments passed to methods
summary	logical indicating if a summary of the colours should be returned as well as the full colour mapping. This will be the unique elements of x mapped to the colour.
n_summaries	positive integer. If supplied a summary colour palette will be returned in a list, containing n_summaries equally spaced values of x in the range [min(x), max(x)], and their associated colours. If a non-numeric x is used this value is ignored
format	logical indicating if the summary values should be formatted. See details
digits	Integer. When summarising a numeric vector you can specify the number of decimal places to include in the summary values

## Details

The palette can either be

- String - "viridis", "inferno", "plasma", "magma", "cividis"
- Matrix - At least 5 rows, and 3 (or 4) columns representing the red, green and blue (and alpha) values

The palette (and alpha argument when using a vector) requires 5 rows because the colours are interpolated using a cubic b-spline. This method requires 5 values.

when `summary = TRUE`, the following rules are applied to the summary values

- logical vectors are converted to "TRUE" or "FALSE" strings
- all other types remain as-is, unless `format = T` is used

when `format = TRUE`,

- numbers are converted to strings with the specified number of decimal places (using digits argument)
- Dates are formatted as "yyyy-mm-dd"

### See Also

colour\_values\_rgb

### Examples

```
## in-built palettes
colour_values(x = 1:5) ## default is "viridis"
colour_values(x = 1:5, palette = "inferno")
colour_values(x = 1:5, palette = "plasma")
colour_values(x = 1:5, palette = "magma")
colour_values(x = 1:5, palette = "cividis")

## matrix palette
n <- 100
m <- grDevices::colorRamp(c("red", "green"))( (1:n)/n )
df <- data.frame(a = 10, x = 1:n)
df$col <- colour_values(df$x, palette = m)
barplot(height = df$a, col = df$col, border = NA, space = 0)

## with an alpha column on the palette
n <- 100
m <- grDevices::colorRamp(c("red", "green"))( (1:n)/n )
m <- cbind(m, seq(0, 255, length.out = 100))
df <- data.frame(a = 10, x = 1:n)
df$col <- colour_values(df$x, palette = m)
barplot(height = df$a, col = df$col, border = NA, space = 0)

## single alpha value for all colours
df <- data.frame(a = 10, x = 1:255)
df$col <- colour_values(df$x, alpha = 50)
barplot(height = df$a, col = df$col, border = NA, space = 0)

## vector of alpha values
df <- data.frame(a = 10, x = 1:300, y = rep(c(1:50, 50:1), 3) )
df$col <- colour_values(df$x, alpha = df$y)
barplot(height = df$a, col = df$col, border = NA, space = 0)

## returning a summary palette
colour_values(-10:10, n_summaries = 5)
```

---

colour_values_rgb	<i>Colour Values RGB</i>
-------------------	--------------------------

---

### Description

Maps colours to variables, returning a matrix of RGB(A) values

### Usage

```
colour_values_rgb(x, palette = "viridis", na_colour = "#808080FF",  
  alpha = 255, include_alpha = TRUE, ...)
```

```
color_values_rgb(x, palette = "viridis", na_colour = "#808080FF",  
  alpha = 255, include_alpha = TRUE, ...)
```

```
## S3 method for class 'character'  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, summary = FALSE)
```

```
## Default S3 method:  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, n_summaries = 0, format = TRUE, digits = 2)
```

```
## S3 method for class 'logical'  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, summary = FALSE)
```

```
## S3 method for class 'factor'  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, summary = FALSE)
```

```
## S3 method for class 'Date'  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, n_summaries = 0, format = TRUE)
```

```
## S3 method for class 'POSIXct'  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, n_summaries = 0, format = TRUE)
```

```
## S3 method for class 'POSIXlt'  
colour_values_to_rgb(x, palette, na_colour, alpha,  
  include_alpha, n_summaries = 0, format = TRUE)
```

### Arguments

x	vector of values to map to a colour
palette	colour palette. See details and examples

na_colour	hex string colour to use for NA values in the form #RRGGBBAA.
alpha	optional. Single value in [0,255] applied to all colours, or a vector of numeric values the same length as x and at least length 5. The numeric vector will be scaled into the range [0,255]. If a matrix palette is supplied this argument is ignored.
include_alpha	logical indicating if the returned hex or matrix should include the alpha values. Defaults to TRUE.
...	other arguments passed to methods
summary	logical indicating if a summary of the colours should be returned as well as the full colour mapping. This will be the unique elements of x mapped to the colour.
n_summaries	positive integer. If supplied a summary colour palette will be returned in a list, containing n_summaries equally spaced values of x in the range [min(x), max(x)], and their associated colours. If a non-numeric x is used this value is ignored
format	logical indicating if the summary values should be formatted. See details
digits	Integer. When summarising a numeric vector you can specify the number of decimal places to include in the summary values

### Details

when `summary = TRUE`, the following rules are applied to the summary values

- logical vectors are converted to "TRUE" or "FALSE" strings
- all other types remain as-is, unless `format = T` is used

when `format = TRUE`,

- numbers are converted to strings with the specified number of decimal places (using `digits` argument)
- Dates are formatted as "yyyy-mm-dd"

### See Also

`colour_values`

### Examples

```
colour_values_rgb(1:5)
colour_values_rgb(1:5, include_alpha = FALSE)
colour_values_rgb(-25:25, n_summaries = 5)
```

---

convert_colour	<i>Convert Colour</i>
----------------	-----------------------

---

**Description**

Converts colours between RRGGBBAA and hex strings, in both directions.

**Usage**

```
convert_colour(x)
```

```
convert_colours(x)
```

```
convert_color(x)
```

```
convert_colors(x)
```

**Arguments**

x                    character vector of hex strings, or numeric matrix of RRGGBBAA values

**Details**

If a combination of hex strings with and without alpha values are supplied, those without are assumed to have an alpha value of FF and will be returned in the RRGGBBAA matrix

**Examples**

```
convert_colour(c("#FFAA00"))
convert_colour(c("#FFAA00", "#FF00A0FF"))

convert_colour(matrix(c(255,170,0),ncol = 3))
convert_colour(matrix(c(255,170,0,255),ncol = 4))
```

---

inferno	<i>Inferno</i>
---------	----------------

---

**Description**

Data frame of the inferno palette

**Usage**

```
inferno()
```

magma

*Magma*

---

**Description**

Data frame of the magma palette

**Usage**

magma()

---

plasma

*Plasma*

---

**Description**

Data frame of the plasma palette

**Usage**

plasma()

---

viridis

*Viridis*

---

**Description**

Data frame of the viridis palette

**Usage**

viridis()



# Index

cividis, 2  
colour\_values (colour\_values), 2  
colour\_values\_rgb (colour\_values\_rgb), 5  
colour\_values, 2  
colour\_values\_rgb, 5  
colour\_values\_to\_hex.character  
    (colour\_values), 2  
colour\_values\_to\_hex.Date  
    (colour\_values), 2  
colour\_values\_to\_hex.default  
    (colour\_values), 2  
colour\_values\_to\_hex.factor  
    (colour\_values), 2  
colour\_values\_to\_hex.logical  
    (colour\_values), 2  
colour\_values\_to\_hex.POSIXct  
    (colour\_values), 2  
colour\_values\_to\_hex.POSIXlt  
    (colour\_values), 2  
colour\_values\_to\_rgb.character  
    (colour\_values\_rgb), 5  
colour\_values\_to\_rgb.Date  
    (colour\_values\_rgb), 5  
colour\_values\_to\_rgb.default  
    (colour\_values\_rgb), 5  
colour\_values\_to\_rgb.factor  
    (colour\_values\_rgb), 5  
colour\_values\_to\_rgb.logical  
    (colour\_values\_rgb), 5  
colour\_values\_to\_rgb.POSIXct  
    (colour\_values\_rgb), 5  
colour\_values\_to\_rgb.POSIXlt  
    (colour\_values\_rgb), 5  
convert\_color (convert\_colour), 7  
convert\_colors (convert\_colour), 7  
convert\_colour, 7  
convert\_colours (convert\_colour), 7  
  
inferno, 7  
  
magma, 8  
plasma, 8  
viridis, 8