

# Package ‘colourvalues’

January 23, 2020

**Type** Package

**Title** Assigns Colours to Values

**Version** 0.3.2

**Date** 2020-01-24

**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, and were set as the default palette for the 'Python' 'Matplotlib' library <<https://matplotlib.org/>>. Other palettes available in this library have been derived from 'RColorBrewer' <<https://CRAN.R-project.org/package=RColorBrewer>> and 'colorspace' <<https://CRAN.R-project.org/package=colorspace>> packages.

**License** GPL-3

**URL** <https://symbolixau.github.io/colourvalues/>

**BugReports** <https://github.com/SymbolixAU/colourvalues/issues>

**Encoding** UTF-8

**LazyData** true

**LinkingTo** Rcpp, BH

**Imports** graphics, Rcpp

**RoxygenNote** 7.0.2

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

**NeedsCompilation** yes

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

**Maintainer** David Cooley <[dcooley@symbolix.com.au](mailto:dcooley@symbolix.com.au)>

**Repository** CRAN

**Date/Publication** 2020-01-23 21:50:02 UTC

## R topics documented:

blue2green . . . . .	3
blue2red . . . . .	3

blue2yellow . . . . .	3
blues . . . . .	4
brbg . . . . .	4
bugn . . . . .	4
bupu . . . . .	4
cividis . . . . .	5
cm . . . . .	5
colour_palettes . . . . .	5
colour_values . . . . .	6
colour_values_rgb . . . . .	9
convert_colour . . . . .	12
cyan2yellow . . . . .	13
diverge_hcl . . . . .	13
diverge_hsv . . . . .	13
get_palette . . . . .	14
gnbu . . . . .	14
green2red . . . . .	15
greens . . . . .	15
greys . . . . .	15
heat . . . . .	15
heat_hcl . . . . .	16
inferno . . . . .	16
magenta2green . . . . .	16
magma . . . . .	16
matlab_like . . . . .	17
matlab_like2 . . . . .	17
oranges . . . . .	17
orrd . . . . .	17
piyg . . . . .	18
plasma . . . . .	18
prgn . . . . .	18
pubu . . . . .	18
pubugn . . . . .	19
puor . . . . .	19
purd . . . . .	19
purples . . . . .	19
rainbow . . . . .	20
rainbow_hcl . . . . .	20
rdbu . . . . .	20
rdgy . . . . .	20
rdpu . . . . .	21
rdylbu . . . . .	21
rdylgn . . . . .	21
reds . . . . .	21
sequential_hcl . . . . .	22
show_colours . . . . .	22
spectral . . . . .	22
terrain . . . . .	23

*blue2green* 3

terrain_hcl . . . . .	23
topo . . . . .	23
viridis . . . . .	23
ygobb . . . . .	24
ylgn . . . . .	24
ylgnbu . . . . .	24
ylorbr . . . . .	24
ylorrd . . . . .	25

**Index** 26

---

*blue2green*                      *Blue2green*

---

**Description**

Data Frame of the *blue2green* palette

**Usage**

`blue2green()`

---

*blue2red*                      *Blue2red*

---

**Description**

Data Frame of the *blue2red* palette

**Usage**

`blue2red()`

---

*blue2yellow*                      *Blue2yellow*

---

**Description**

Data Frame of the *blue2yellow* palette

**Usage**

`blue2yellow()`

blues *Blues*

---

**Description**

Data Frame of the blues palette

**Usage**

blues()

---

brbg *Brbg*

---

**Description**

Data Frame of the brbg palette

**Usage**

brbg()

---

bugn *Bugn*

---

**Description**

Data Frame of the bugn palette

**Usage**

bugn()

---

bupu *Bupu*

---

**Description**

Data Frame of the bupu palette

**Usage**

bupu()

---

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

---

**Description**

Data frame of the cividis palette

**Usage**

```
cividis()
```

---

cm	<i>Cm</i>
----	-----------

---

**Description**

Data Frame of the cm palette

**Usage**

```
cm()
```

---

colour_palettes	<i>Colour Palettes</i>
-----------------	------------------------

---

**Description**

List the available colour palettes.

**Usage**

```
colour_palettes(colours = NULL)
```

```
color_palettes(colours = NULL)
```

**Arguments**

colours	vector of source colour palettes to return, one or many of "viridis", "rcolorbrewer", "grdevices", "colorspace" NULL will reutrn all palettes.
---------	---

## Details

The palettes available in colourvalues have been derived from those available in the libraries

- viridis
- RColorBrewer
- grDevices
- colorspace
- colorRamp

## Examples

```
colour_palettes()  
colour_palettes( "viridis" )  
colour_palettes( colours = c("rcolorbrewer", "grdevices") )
```

---

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

---

## Description

maps colours to values

## Usage

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

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

```
## S3 method for class 'character'  
colour_values_to_hex(  
  x,
```

```
    palette,  
    alpha,  
    na_colour,  
    include_alpha,  
    summary = FALSE  
  )  
  
## S3 method for class 'logical'  
colour_values_to_hex(  
  x,  
  palette,  
  alpha,  
  na_colour,  
  include_alpha,  
  summary = FALSE  
)  
  
## S3 method for class 'factor'  
colour_values_to_hex(  
  x,  
  palette,  
  alpha,  
  na_colour,  
  include_alpha,  
  summary = FALSE  
)  
  
## S3 method for class 'Date'  
colour_values_to_hex(  
  x,  
  palette,  
  alpha,  
  na_colour,  
  include_alpha,  
  n_summaries = 0,  
  format = TRUE  
)  
  
## S3 method for class 'POSIXct'  
colour_values_to_hex(  
  x,  
  palette,  
  alpha,  
  na_colour,  
  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

<code>x</code>	vector of values to map to a colour
<code>palette</code>	colour palette. See details and examples
<code>alpha</code>	optional. Single value in [0,255] applied to all colours, or a decimal in [0, 1) (to indicate a percentage, noting 1 is excluded), or a vector of numeric values the same length as <code>x</code> . The numeric vector will be scaled into the range [0,255]. If a matrix palette is supplied this argument is ignored.
<code>na_colour</code>	hex string colour to use for NA values in the form #RRGGBBAA.
<code>include_alpha</code>	logical indicating if the returned hex or matrix should include the alpha values. Defaults to TRUE.
<code>...</code>	other arguments passed to methods
<code>summary</code>	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 <code>x</code> mapped to the colour.
<code>n_summaries</code>	positive integer. If supplied a summary colour palette will be returned in a list, containing <code>n_summaries</code> equally spaced values of <code>x</code> in the range $[\min(x), \max(x)]$ , and their associated colours. If a non-numeric <code>x</code> is used this value is ignored
<code>format</code>	logical indicating if the summary values should be formatted. See details

### Details

The palette can either be

- String - use `colour_palettes()` to view available palettes
- Matrix - At least 5 rows, and 3 (or 4) columns representing the red, green and blue (and alpha) values

The matrix palette requires 5 rows because the colours are interpolated using a cubic b-spline. This method requires 5 values.

### 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")
colour_values(x = 1:5, palette = "rainbow")

## 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

*Colour Values RGB*


---

**Description**

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

**Usage**

```

colour_values_rgb(
  x,

```

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

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

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

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

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

## S3 method for class 'Date'
colour_values_to_rgb(
```

```

    x,
    palette,
    alpha,
    na_colour,
    include_alpha,
    n_summaries = 0,
    format = TRUE
)

## S3 method for class 'POSIXct'
colour_values_to_rgb(
  x,
  palette,
  alpha,
  na_colour,
  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

<code>x</code>	vector of values to map to a colour
<code>palette</code>	colour palette. See details and examples
<code>alpha</code>	optional. Single value in [0,255] applied to all colours, or a decimal in [0, 1) (to indicate a percentage, noting 1 is excluded), or a vector of numeric values the same length as <code>x</code> . The numeric vector will be scaled into the range [0,255]. If a matrix palette is supplied this argument is ignored.
<code>na_colour</code>	hex string colour to use for NA values in the form #RRGGBBAA.
<code>include_alpha</code>	logical indicating if the returned hex or matrix should include the alpha values. Defaults to TRUE.
<code>...</code>	other arguments passed to methods
<code>summary</code>	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 <code>x</code> mapped to the colour.
<code>n_summaries</code>	positive integer. If supplied a summary colour palette will be returned in a list, containing <code>n_summaries</code> equally spaced values of <code>x</code> 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

### Details

The palette can either be

- String - use `colour_palettes()` to view available palettes
- Matrix - At least 5 rows, and 3 (or 4) columns representing the red, green and blue (and alpha) values

The matrix palette requires 5 rows because the colours are interpolated using a cubic b-spline. This method requires 5 values.

### 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

*Convert Colour*

---

### 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))
```

---

cyan2yellow	<i>Cyan2yellow</i>
-------------	--------------------

---

**Description**

Data Frame of the cyan2yellow palette

**Usage**

```
cyan2yellow()
```

---

diverge_hcl	<i>Diverge_hcl</i>
-------------	--------------------

---

**Description**

Data Frame of the diverge\_hcl palette

**Usage**

```
diverge_hcl()
```

---

diverge_hsv	<i>Diverge_hsv</i>
-------------	--------------------

---

**Description**

Data Frame of the diverge\_hsv palette

**Usage**

```
diverge_hsv()
```

`get_palette`*Get Palette*

---

**Description**

retrieves one of the available palettes

**Usage**

```
get_palette(palette, rgb = TRUE)
```

**Arguments**

`palette` one of the available palettes. See [colour\\_palettes](#)  
`rgb` logical indicating if the palette should be returned as an RGB matrix TRUE, or a vector of hex strings FALSE

**Value**

256 row x 3 column matrix if `rgb = TRUE`, otherwise a 256-length vector.

**Examples**

```
get_palette( "viridis" )  
get_palette( "rainbow" )
```

---

`gnbu`*Gnbu*

---

**Description**

Data Frame of the gnbu palette

**Usage**

```
gnbu()
```

---

green2red	<i>Green2red</i>
-----------	------------------

---

**Description**

Data Frame of the green2red palette

**Usage**

green2red()

---

greens	<i>Greens</i>
--------	---------------

---

**Description**

Data Frame of the greens palette

**Usage**

greens()

---

greys	<i>Greys</i>
-------	--------------

---

**Description**

Data Frame of the greys palette

**Usage**

greys()

---

heat	<i>Heat</i>
------	-------------

---

**Description**

Data Frame of the heat palette

**Usage**

heat()

---

heat_hcl	<i>Heat_hcl</i>
----------	-----------------

---

**Description**

Data Frame of the heat\_hcl palette

**Usage**

```
heat_hcl()
```

---

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

---

**Description**

Data frame of the inferno palette

**Usage**

```
inferno()
```

---

magenta2green	<i>Magenta2green</i>
---------------	----------------------

---

**Description**

Data Frame of the magenta2green palette

**Usage**

```
magenta2green()
```

---

magma	<i>Magma</i>
-------	--------------

---

**Description**

Data frame of the magma palette

**Usage**

```
magma()
```



---

matlab_like	<i>Matlab_like</i>
-------------	--------------------

---

**Description**

Data Frame of the matlab\_like palette

**Usage**

matlab\_like()

---

matlab_like2	<i>Matlab_like2</i>
--------------	---------------------

---

**Description**

Data Frame of the matlab\_like2 palette

**Usage**

matlab\_like2()

---

oranges	<i>Oranges</i>
---------	----------------

---

**Description**

Data Frame of the oranges palette

**Usage**

oranges()

---

orrd	<i>Orrd</i>
------	-------------

---

**Description**

Data Frame of the orrd palette

**Usage**

orrd()

---

piyg	<i>Piyg</i>
------	-------------

---

**Description**

Data Frame of the piyg palette

**Usage**

piyg()

---

plasma	<i>Plasma</i>
--------	---------------

---

**Description**

Data frame of the plasma palette

**Usage**

plasma()

---

prgn	<i>Prgn</i>
------	-------------

---

**Description**

Data Frame of the prgn palette

**Usage**

prgn()

---

pubu	<i>Pubu</i>
------	-------------

---

**Description**

Data Frame of the pubu palette

**Usage**

pubu()

---

pubugn	<i>Pubugn</i>
--------	---------------

---

**Description**

Data Frame of the pubugn palette

**Usage**

pubugn()

---

puor	<i>Puor</i>
------	-------------

---

**Description**

Data Frame of the puor palette

**Usage**

puor()

---

purd	<i>Purd</i>
------	-------------

---

**Description**

Data Frame of the purd palette

**Usage**

purd()

---

purples	<i>Purples</i>
---------	----------------

---

**Description**

Data Frame of the purples palette

**Usage**

purples()

---

rainbow	<i>Rainbow</i>
---------	----------------

---

**Description**

Data Frame of the rainbow palette

**Usage**

```
rainbow()
```

---

rainbow_hcl	<i>Rainbow_hcl</i>
-------------	--------------------

---

**Description**

Data Frame of the rainbow\_hcl palette

**Usage**

```
rainbow_hcl()
```

---

rdbu	<i>Rdbu</i>
------	-------------

---

**Description**

Data Frame of the rdbu palette

**Usage**

```
rdbu()
```

---

rdgy	<i>Rdgy</i>
------	-------------

---

**Description**

Data Frame of the rdgy palette

**Usage**

```
rdgy()
```

---

rdpu	<i>Rdpu</i>
------	-------------

---

**Description**

Data Frame of the rdpu palette

**Usage**

rdpu()

---

rdylbu	<i>Rdylbu</i>
--------	---------------

---

**Description**

Data Frame of the rdylbu palette

**Usage**

rdylbu()

---

rdylgn	<i>Rdylgn</i>
--------	---------------

---

**Description**

Data Frame of the rdylgn palette

**Usage**

rdylgn()

---

reds	<i>Reds</i>
------	-------------

---

**Description**

Data Frame of the reds palette

**Usage**

reds()

---

sequential_hcl	<i>Sequential_hcl</i>
----------------	-----------------------

---

**Description**

Data Frame of the sequential\_hcl palette

**Usage**

```
sequential_hcl()
```

---

show_colours	<i>Show Colours</i>
--------------	---------------------

---

**Description**

Plots all the selected colours. See [colour\\_palettes](#) for available colours.

**Usage**

```
show_colours(colours = colour_palettes())
```

**Arguments**

colours            vector of colour palettes

**Examples**

```
## view all the colour palettes
show_colours()

## view a selection of colour palettes
show_colours( colours = colour_palettes( c("viridis", "grdevices") ) )
```

---

spectral	<i>Spectral</i>
----------	-----------------

---

**Description**

Data Frame of the spectral palette

**Usage**

```
spectral()
```

---

terrain	<i>Terrain</i>
---------	----------------

---

**Description**

Data frame of the terrain palette

**Usage**

```
terrain()
```

---

terrain_hcl	<i>Terrain_hcl</i>
-------------	--------------------

---

**Description**

Data Frame of the terrain\_hcl palette

**Usage**

```
terrain_hcl()
```

---

topo	<i>Topo</i>
------	-------------

---

**Description**

Data Frame of the topo palette

**Usage**

```
topo()
```

---

viridis	<i>Viridis</i>
---------	----------------

---

**Description**

Data frame of the viridis palette

**Usage**

```
viridis()
```

---

ygobb	<i>Ygobb</i>
-------	--------------

---

**Description**

Data Frame of the ygobb palette

**Usage**

ygobb()

---

ylgn	<i>Ylgn</i>
------	-------------

---

**Description**

Data Frame of the ylgn palette

**Usage**

ylgn()

---

ylgnbu	<i>Ylgnbu</i>
--------	---------------

---

**Description**

Data Frame of the ylgnbu palette

**Usage**

ylgnbu()

---

ylorbr	<i>Ylorbr</i>
--------	---------------

---

**Description**

Data Frame of the ylorbr palette

**Usage**

ylorbr()



---

<code>ylorrd</code>	<i>Ylorrd</i>
---------------------	---------------

---

**Description**

Data Frame of the ylorrd palette

**Usage**

`ylorrd()`

# Index

blue2green, 3  
blue2red, 3  
blue2yellow, 3  
blues, 4  
brbg, 4  
bugn, 4  
bupu, 4

cividis, 5  
cm, 5  
color\_palettes (colour\_palettes), 5  
color\_values (colour\_values), 6  
color\_values\_rgb (colour\_values\_rgb), 9  
colour\_palettes, 5, 14, 22  
colour\_values, 6  
colour\_values\_rgb, 9  
colour\_values\_to\_hex.character  
    (colour\_values), 6  
colour\_values\_to\_hex.Date  
    (colour\_values), 6  
colour\_values\_to\_hex.factor  
    (colour\_values), 6  
colour\_values\_to\_hex.logical  
    (colour\_values), 6  
colour\_values\_to\_hex.POSIXct  
    (colour\_values), 6  
colour\_values\_to\_hex.POSIXlt  
    (colour\_values), 6  
colour\_values\_to\_rgb.character  
    (colour\_values\_rgb), 9  
colour\_values\_to\_rgb.Date  
    (colour\_values\_rgb), 9  
colour\_values\_to\_rgb.factor  
    (colour\_values\_rgb), 9  
colour\_values\_to\_rgb.logical  
    (colour\_values\_rgb), 9  
colour\_values\_to\_rgb.POSIXct  
    (colour\_values\_rgb), 9  
colour\_values\_to\_rgb.POSIXlt  
    (colour\_values\_rgb), 9

convert\_color (convert\_colour), 12  
convert\_colors (convert\_colour), 12  
convert\_colour, 12  
convert\_colours (convert\_colour), 12  
cyan2yellow, 13

diverge\_hcl, 13  
diverge\_hsv, 13

get\_palette, 14  
gnbu, 14  
green2red, 15  
greens, 15  
greys, 15

heat, 15  
heat\_hcl, 16

inferno, 16

magenta2green, 16  
magma, 16  
matlab\_like, 17  
matlab\_like2, 17

oranges, 17  
orrd, 17

piyg, 18  
plasma, 18  
prgn, 18  
pubu, 18  
pubugn, 19  
puor, 19  
purd, 19  
purples, 19

rainbow, 20  
rainbow\_hcl, 20  
rdbu, 20  
rdgy, 20

rdpu, 21

rdylbu, 21

rdylgn, 21

reds, 21

sequential\_hcl, 22

show\_colours, 22

spectral, 22

terrain, 23

terrain\_hcl, 23

topo, 23

viridis, 23

ygobb, 24

ylgn, 24

ylgnbu, 24

ylorbr, 24

ylorrd, 25