

Package ‘khroma’

February 25, 2025

Title Colour Schemes for Scientific Data Visualization

Version 1.16.0

Description Color schemes ready for each type of data (qualitative, diverging or sequential), with colors that are distinct for all people, including color-blind readers. This package provides an implementation of Paul Tol (2018) and Fabio Crameri (2018) <doi:10.5194/gmd-11-2541-2018> color schemes for use with 'graphics' or 'ggplot2'. It provides tools to simulate color-blindness and to test how well the colors of any palette are identifiable. Several scientific thematic schemes (geologic timescale, land cover, FAO soils, etc.) are also implemented.

License GPL (>= 3)

URL <https://codeberg.org/tesselle/khroma>,
<https://packages.tesselle.org/khroma/>

BugReports <https://codeberg.org/tesselle/khroma/issues>

Depends R (>= 3.5.0)

Imports graphics, grDevices, grid, stats, utils

Suggests ggplot2, ggraph, knitr, markdown, rsvg, scales, spacesXYZ, svglite, tinysnapshot, tinytest

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.2

X-schema.org-isPartOf <https://www.tesselle.org>

X-schema.org-keywords data-visualization, colour-schemes, accessibility, r-package

Collate 'change.R' 'color.R' 'compare.R' 'info.R' 'khroma-defunct.R' 'khroma-deprecated.R' 'khroma-internal.R' 'khroma-package.R' 'palettes.R' 'plot.R' 'plot_map.R' 'plot_scheme.R' 'plot_scheme_colorblind.R' 'plot_tiles.R' 'print.R' 'scale_colour_crameri.R' 'scale_colour_okabeito.R' 'scale_colour_picker.R' 'scale_colour_science.R' 'scale_colour_tol.R'

NeedsCompilation no

Author Nicolas Frerebeau [aut, cre] (<<https://orcid.org/0000-0001-5759-4944>>),
 Brice Lebrun [art] (<<https://orcid.org/0000-0001-7503-8685>>, Logo
 designer),
 Vincent Arel-Bundock [ctb] (<<https://orcid.org/0000-0003-2042-7063>>),
 Ulrik Stervbo [ctb] (<<https://orcid.org/0000-0002-2831-8868>>),
 Université Bordeaux Montaigne [fnd] (03pbgwk21),
 CNRS [fnd] (02feahw73)

Maintainer Nicolas Frerebeau <nicolas.frerebeau@u-bordeaux-montaigne.fr>

Repository CRAN

Date/Publication 2025-02-25 17:40:05 UTC

Contents

change	4
colour	5
compare	8
info	9
palette_color_continuous	10
palette_color_discrete	11
palette_color_picker	13
palette_shape	14
palette_size	15
plot.color_scheme	17
plot_map	18
plot_scheme	19
plot_scheme_colourblind	20
plot_tiles	21
scale_colour_land	22
scale_colour_soil	24
scale_colour_stratigraphy	25
scale_crameri_acton	27
scale_crameri_bam	30
scale_crameri_bamako	33
scale_crameri_bamO	36
scale_crameri_batlow	39
scale_crameri_batlowK	42
scale_crameri_batlowW	45
scale_crameri_berlin	48
scale_crameri_bilbao	51
scale_crameri_broc	54
scale_crameri_brocO	57
scale_crameri_buda	60
scale_crameri_bukavu	63
scale_crameri_cork	65
scale_crameri_corkO	68

scale_crameri_davos 71

scale_crameri_devon 74

scale_crameri_fes 77

scale_crameri_grayC 80

scale_crameri_hawaii 83

scale_crameri_imola 86

scale_crameri_lajolla 89

scale_crameri_lapaz 92

scale_crameri_lisbon 95

scale_crameri_nuuk 98

scale_crameri_oleron 102

scale_crameri_oslo 104

scale_crameri_roma 107

scale_crameri_romaO 110

scale_crameri_tofino 113

scale_crameri_tokyo 116

scale_crameri_turku 119

scale_crameri_vanimo 122

scale_crameri_vik 125

scale_crameri_vikO 128

scale_okabeito_discrete 130

scale_picker 132

scale_tol_bright 133

scale_tol_BuRd 135

scale_tol_dark 138

scale_tol_discreterainbow 140

scale_tol_highcontrast 142

scale_tol_incandescent 145

scale_tol_iridescent 147

scale_tol_light 150

scale_tol_mediumcontrast 152

scale_tol_muted 154

scale_tol_nightfall 156

scale_tol_pale 159

scale_tol_PRGn 161

scale_tol_smoothrainbow 163

scale_tol_sunset 166

scale_tol_vibrant 169

scale_tol_YlOrBr 171

change

Simulate Color-Blindness

Description

Simulate Color-Blindness

Usage

```
change(x, mode)
```

Arguments

x	A palette function that when called with a single integer argument (the number of levels) returns a vector of colors (see color()).
mode	A character string giving the colorblind vision to be used. It must be one of "deuteranopia", "protanopia", "tritanopia" or "achromatopsia". Any unambiguous substring can be given.

Value

A palette [function](#) that returns a vector of anomalized colors. All the attributes of the initial palette function are inherited, with a supplementary attribute "mode" giving the corresponding color-blind vision.

Author(s)

N. Frerebeau

References

Brettel, H., Viénot, F. and Mollon, J. D. (1997). Computerized Simulation of Color Appearance for Dichromats. *Journal of the Optical Society of America A*, 14(10), p. 2647-2655. doi:10.1364/JOSAA.14.002647.

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1. URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

Viénot, F., Brettel, H. and Mollon, J. D. (1999). Digital Video Colourmaps for Checking the Legibility of Displays by Dichromats. *Color Research & Application*, 24(4), p. 243-52. doi:10.1002/(SICI)15206378(199908)24:4<243::AIDCOL5>3.0.CO;23.

See Also

Other diagnostic tools: [compare\(\)](#), [plot.color_scheme\(\)](#), [plot_map\(\)](#), [plot_scheme\(\)](#), [plot_scheme_colourblind\(\)](#), [plot_tiles\(\)](#)

Examples

```
if (requireNamespace("spacesXYZ", quietly = TRUE)) {
  ## Trichromat
  pal <- colour("bright")
  plot_scheme(pal(7))

  ## Deuteranopia
  deu <- change(pal, mode = "deuteranopia")
  plot_scheme(deu(7))

  ## Protanopia
  pro <- change(pal, mode = "protanopia")
  plot_scheme(pro(7))

  ## Tritanopia
  tri <- change(pal, mode = "tritanopia")
  plot_scheme(tri(7))

  ## Achromatopsia
  ach <- change(pal, mode = "achromatopsia")
  plot_scheme(ach(7))

  ## Plot simulated color blindness
  plot_scheme_colorblind(pal(7))
}
```

colour

Color Schemes

Description

Provides qualitative, diverging and sequential color schemes.

Usage

```
colour(
  palette,
  reverse = FALSE,
  names = FALSE,
  lang = "en",
  force = FALSE,
  ...
)
```

```
color(palette, reverse = FALSE, names = FALSE, lang = "en", force = FALSE, ...)
```

Arguments

palette	A character string giving the name of the scheme to be used (see info()).
reverse	A logical scalar: should the resulting vector of colors should be reversed?
names	A logical scalar: should the names of the colors should be kept in the resulting vector?
lang	A character string specifying the language for the color names. It must be one of "en" (English, the default) or "fr" (French).
force	A logical scalar. If TRUE, forces the color scheme to be interpolated. It should not be used routinely with qualitative color schemes, as they are designed to be used as is to remain color-blind safe.
...	Further arguments passed to colorRampPalette .

Value

A [function](#) function with the following attributes, that when called with a single argument (an [integer](#) specifying the number of colors) returns a (named) vector of colors.

palette A [character](#) string giving the name of the color scheme.

type A [character](#) string giving the corresponding data type. One of "qualitative", "diverging" or "sequential".

interpolate A [logical](#) scalar: can the color palette be interpolated?

missing A [character](#) string giving the the hexadecimal representation of the color that should be used for NA values.

max An [integer](#) giving the maximum number of color values. Only relevant for non-interpolated color schemes.

For color schemes that can be interpolated (diverging and sequential data), the color range can be limited with an additional argument. `range` allows to remove a fraction of the color domain (before being interpolated; see examples).

Author(s)

N. Frerebeau

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607
- Jones, A., Montanarella, L. & Jones, R. (Ed.) (2005). *Soil atlas of Europe*. Luxembourg: European Commission, Office for Official Publications of the European Communities. 128 pp. ISBN: 92-894-8120-X.
- Okabe, M. & Ito, K. (2008). *Color Universal Design (CUD): How to Make Figures and Presentations That Are Friendly to Colorblind People*. URL: <https://jfly.uni-koeln.de/color/>.

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

Commission for the Geological Map of the World

See Also

Other color schemes: [info\(\)](#)

Examples

```
## Okabe and Ito colour scheme
colour("okabe ito")(8)
plot_scheme(colour("okabe ito")(8))

## Paul Tol's colour schemes
### Qualitative data
plot_scheme(colour("bright")(7))
plot_scheme(colour("high contrast")(3))
plot_scheme(colour("vibrant")(7))
plot_scheme(colour("muted")(9))
plot_scheme(colour("medium contrast")(6))
plot_scheme(colour("pale")(6))
plot_scheme(colour("dark")(6))
plot_scheme(colour("light")(9))
### Diverging data
plot_scheme(colour("sunset")(11))
plot_scheme(colour("BuRd")(9))
plot_scheme(colour("PRGn")(9))
### Sequential data
plot_scheme(colour("YlOrBr")(9))
plot_scheme(colour("iridescent")(23))
plot_scheme(colour("discrete rainbow")(14))
plot_scheme(colour("discrete rainbow")(23))
plot_scheme(colour("smooth rainbow")(34))

## Scientific colour schemes
### Geologic timescale
plot_scheme(colour("stratigraphy")(175))
### AVHRR global land cover classification
plot_scheme(colour("land")(14))
### FAO soil reference groups
plot_scheme(colour("soil")(24))

## Adjust colour levels
PRGn <- colour("PRGn")
plot_scheme(PRGn(9, range = c(0.5, 1)))
```

compare

Color Difference

Description

Computes CIELAB distance metric.

Usage

```
compare(x, metric = 2000, diag = FALSE, upper = FALSE)
```

Arguments

x	A character vector of colors.
metric	An integer value giving the year the metric was recommended by the CIE. It must be one of "1976", "1994", or "2000" (default; see spacesXYZ::DeltaE()).
diag	A logical scalar: should the diagonal of the distance matrix be printed?
upper	A logical scalar: should the upper triangle of the distance matrix should be printed?

Value

A [distance matrix](#).

Author(s)

N. Frerebeau

See Also

Other diagnostic tools: [change\(\)](#), [plot.color_scheme\(\)](#), [plot_map\(\)](#), [plot_scheme\(\)](#), [plot_scheme_colourblind\(\)](#), [plot_tiles\(\)](#)

Examples

```
if (requireNamespace("spacesXYZ", quietly = TRUE)) {  
  ## Trichromat  
  pal <- colour("bright")  
  compare(pal(5))  
  
  ## Deuteranopia  
  deu <- change(pal, mode = "deuteranopia")  
  compare(deu(5))  
  
  ## Protanopia  
  pro <- change(pal, mode = "protanopia")  
  compare(pro(5))  
}
```



```
## Tritanopia
tri <- change(pal, mode = "tritanopia")
compare(tri(5))

## Achromatopsia
ach <- change(pal, mode = "achromatopsia")
compare(ach(5))
}
```

info

Available Schemes

Description

Returns information about the available schemes.

Usage

```
info()
```

Value

A [data.frame](#) with the following columns:

palette Names of palette.

type Types of schemes: sequential, diverging or qualitative.

max Maximum number of colors that are contained in each palette. Only relevant for qualitative schemes.

missing The hexadecimal color value for mapping missing values.

Author(s)

N. Frerebeau

See Also

Other color schemes: [colour\(\)](#)

Examples

```
## Get a table of available palettes
info()
```

`palette_color_continuous`*Color Mapping (continuous)*

Description

Maps continuous values to an interpolated colors gradient.

Usage

```
palette_color_continuous(  
  colors = NULL,  
  domain = NULL,  
  midpoint = NULL,  
  missing = "#DDDDDD"  
)  
  
palette_colour_continuous(  
  colors = NULL,  
  domain = NULL,  
  midpoint = NULL,  
  missing = "#DDDDDD"  
)
```

Arguments

<code>colors</code>	A vector of colors or a function that when called with a single argument (an integer specifying the number of colors) returns a vector of colors. If NULL (the default), uses <i>YlOrRd</i> .
<code>domain</code>	A numeric range specifying the possible values that can be mapped.
<code>midpoint</code>	A length-one numeric vector specifying the mid-point of input range.
<code>missing</code>	The color to return for NA values.

Value

A palette [function](#) that when called with a single argument (a [numeric](#) vector of continuous values) returns a [character](#) vector of colors.

See Also

[grDevices::colorRamp\(\)](#)

Other palettes: [palette_color_discrete\(\)](#), [palette_color_picker\(\)](#), [palette_shape\(\)](#), [palette_size](#)

Examples

```
## Visualize a simple DEM model
## Distribution of elevation values
elevation <- hist(volcano)

## Where are breaks?
elevation$breaks

## Build palette functions
BuRd <- color("BuRd")
ramp_BuRd <- palette_color_continuous(colors = BuRd(10))

## Plot image
image(volcano, col = ramp_BuRd(elevation$breaks))
legend("topright", legend = elevation$breaks, fill = ramp_BuRd(elevation$breaks))

## Scatter plot
## Build color palette functions
YlOrBr <- color("YlOrBr")
pal_color <- palette_color_continuous(colors = YlOrBr)

## Build symbol palette functions
pal_size <- palette_size_sequential(range = c(1, 3))

## Plot
plot(
  x = iris$Petal.Length,
  y = iris$Sepal.Length,
  pch = 16,
  col = pal_color(iris$Petal.Length),
  cex = pal_size(iris$Petal.Length),
  xlab = "Petal length",
  ylab = "Sepal length",
  panel.first = grid(),
  las = 1
)
```

palette_color_discrete

Color Mapping (discrete)

Description

Maps categorical values to colors.

Usage

```
palette_color_discrete(
  colors = NULL,
```

```

    domain = NULL,
    ordered = FALSE,
    missing = "#DDDDDD"
  )

palette_colour_discrete(
  colors = NULL,
  domain = NULL,
  ordered = FALSE,
  missing = "#DDDDDD"
)

```

Arguments

colors	A vector of colors or a function that when called with a single argument (an integer specifying the number of colors) returns a vector of colors. If NULL (the default), uses <i>discrete rainbow</i> .
domain	A vector of categorical data specifying the possible values that can be mapped.
ordered	A logical scalar: should the levels be treated as already in the correct order?
missing	The color to return for NA values.

Value

A palette [function](#) that when called with a single argument (a vector of categorical values) returns a [character](#) vector of colors.

See Also

Other palettes: [palette_color_continuous\(\)](#), [palette_color_picker\(\)](#), [palette_shape\(\)](#), [palette_size](#)

Examples

```

## Scatter plot
## Build color palette functions
bright <- c(versicolor = "#4477AA", virginica = "#EE6677", setosa = "#228833")
pal_color <- palette_color_discrete(colors = bright)

## Build symbol palette functions
symbols <- c(versicolor = 15, virginica = 16, setosa = 17)
pal_shapes <- palette_shape(symbols)

## Plot
plot(
  x = iris$Petal.Length,
  y = iris$Sepal.Length,
  pch = pal_shapes(iris$Species),
  col = pal_color(iris$Species),
  xlab = "Petal length",
  ylab = "Sepal length",

```

```
  panel.first = grid(),
  las = 1
)
legend("topleft", legend = names(bright), col = bright, pch = symbols)
```

palette_color_picker *Color Mapping*

Description

Maps values to colors.

Usage

```
palette_color_picker(
  scheme,
  domain = NULL,
  midpoint = NULL,
  ordered = FALSE,
  missing = NULL,
  ...
)
```

```
palette_colour_picker(
  scheme,
  domain = NULL,
  midpoint = NULL,
  ordered = FALSE,
  missing = NULL,
  ...
)
```

Arguments

scheme	A character string giving the name of the scheme to be used (see color()).
domain	A numeric range or a vector of categorical data specifying the possible values that can be mapped.
midpoint	A length-one numeric vector specifying the mid-point of input range.
ordered	A logical scalar: should the levels be treated as already in the correct order?
missing	The color to return for NA values.
...	Further parameters to be passed to color() .

Details

A wrapper around [palette_color_continuous\(\)](#) and [palette_color_discrete\(\)](#).

Value

A palette [function](#) that when called with a single argument returns a [character](#) vector of colors.

See Also

Other palettes: [palette_color_continuous\(\)](#), [palette_color_discrete\(\)](#), [palette_shape\(\)](#), [palette_size](#)

Examples

```
## Visualize a simple DEM model
## Distribution of elevation values
elevation <- hist(volcano)

## Where are breaks?
elevation$breaks

## Build palette functions
ramp_BuRd <- palette_color_picker("BuRd")

(col <- ramp_BuRd(elevation$breaks))
image(volcano, col = col)
legend("topright", legend = elevation$breaks, fill = col)

## Rescale to midpoint
ramp_BuRd <- palette_color_picker("BuRd", midpoint = 160)

(col <- ramp_BuRd(elevation$breaks))
image(volcano, col = col)
legend("topright", legend = elevation$breaks, fill = col)
```

palette_shape

Symbol Mapping

Description

Symbol Mapping

Usage

```
palette_shape(symbols = NULL, domain = NULL, ordered = FALSE, ...)
```

```
palette_line(types = NULL, domain = NULL, ordered = FALSE, ...)
```

Arguments

symbols, types	A vector of symbols or line types.
domain	A vector of categorical data specifying the possible values that can be mapped.
ordered	A logical scalar: should the levels be treated as already in the correct order?
...	Currently not used.

Value

A palette [function](#) that when called with a single argument (a [character](#) vector of categorical values) returns a vector of symbols.

See Also

Other palettes: [palette_color_continuous\(\)](#), [palette_color_discrete\(\)](#), [palette_color_picker\(\)](#), [palette_size](#)

Examples

```
## Scatter plot
## Build color palette functions
bright <- c(versicolor = "#4477AA", virginica = "#EE6677", setosa = "#228833")
pal_color <- palette_color_discrete(colors = bright)

## Build symbol palette functions
symbols <- c(versicolor = 15, virginica = 16, setosa = 17)
pal_shapes <- palette_shape(symbols)

## Plot
plot(
  x = iris$Petal.Length,
  y = iris$Sepal.Length,
  pch = pal_shapes(iris$Species),
  col = pal_color(iris$Species),
  xlab = "Petal length",
  ylab = "Sepal length",
  panel.first = grid(),
  las = 1
)
legend("topleft", legend = names(bright), col = bright, pch = symbols)
```

palette_size

Symbol Size Mapping

Description

Symbol Size Mapping

Usage

```
palette_size_sequential(range = c(1, 6), ...)
```

```
palette_size_diverging(range = c(1, 6), midpoint = 0, ...)
```

Arguments

range	A length-two numeric vector giving range of possible sizes (greater than 0).
...	Currently not used.
midpoint	A length-one numeric vector specifying the data mid-point.

Value

A palette **function** that when called with a single argument (a **numeric** vector of continuous values) returns a **numeric** vector giving the amount by which plotting text and symbols should be magnified relative to the default.

See Also

Other palettes: [palette_color_continuous\(\)](#), [palette_color_discrete\(\)](#), [palette_color_picker\(\)](#), [palette_shape\(\)](#)

Examples

```
## Visualize a simple DEM model
## Distribution of elevation values
elevation <- hist(volcano)

## Where are breaks?
elevation$breaks

## Build palette functions
BuRd <- color("BuRd")
ramp_BuRd <- palette_color_continuous(colors = BuRd(10))

## Plot image
image(volcano, col = ramp_BuRd(elevation$breaks))
legend("topright", legend = elevation$breaks, fill = ramp_BuRd(elevation$breaks))

## Scatter plot
## Build color palette functions
YlOrBr <- color("YlOrBr")
pal_color <- palette_color_continuous(colors = YlOrBr)

## Build symbol palette functions
pal_size <- palette_size_sequential(range = c(1, 3))

## Plot
plot(
  x = iris$Petal.Length,
  y = iris$Sepal.Length,
  pch = 16,
  col = pal_color(iris$Petal.Length),
  cex = pal_size(iris$Petal.Length),
  xlab = "Petal length",
  ylab = "Sepal length",
  panel.first = grid(),
```



```
  las = 1
)
```

plot.color_scheme *Plot Color Scheme*

Description

Quickly displays a color scheme returned by [color\(\)](#).

Usage

```
## S3 method for class 'color_scheme'
plot(x, ...)
```

Arguments

x A [character](#) vector of colors.
... Currently not used.

Value

plot() is called for its side-effects: it results in a graphic being displayed (invisibly returns x).

Author(s)

N. Frerebeau

See Also

Other diagnostic tools: [change\(\)](#), [compare\(\)](#), [plot_map\(\)](#), [plot_scheme\(\)](#), [plot_scheme_colourblind\(\)](#), [plot_tiles\(\)](#)

Examples

```
plot(colour("bright")(7))
plot(colour("smooth rainbow")(256))

## Plot colour schemes
plot_scheme(colour("bright")(7))
plot_scheme(colour("sunset")(11))
plot_scheme(colour("YlOrBr")(9))
plot_scheme(colour("discrete rainbow")(14))

## Plot diagnostic maps
plot_map(colour("bright")(7))
plot_map(colour("sunset")(11))
plot_map(colour("YlOrBr")(9))
plot_map(colour("discrete rainbow")(14))
```

```
## Plot diagnostic images
plot_tiles(colour("discrete rainbow")(14), n = 256)
plot_tiles(colour("discrete rainbow")(23), n = 256)
plot_tiles(colour("smooth rainbow")(256), n = 256)
```

plot_map

Diagnostic Map

Description

Produces a diagnostic map for a given color scheme.

Usage

```
plot_map(x)
```

Arguments

x A [character](#) vector of colors.

Value

plot_map() is called for its side-effects: it results in a graphic being displayed (invisibly returns x).

Author(s)

N. Frerebeau, V. Arel-Bundock

See Also

Other diagnostic tools: [change\(\)](#), [compare\(\)](#), [plot.color_scheme\(\)](#), [plot_scheme\(\)](#), [plot_scheme_colourblind\(\)](#), [plot_tiles\(\)](#)

Examples

```
plot(colour("bright")(7))
plot(colour("smooth rainbow")(256))

## Plot colour schemes
plot_scheme(colour("bright")(7))
plot_scheme(colour("sunset")(11))
plot_scheme(colour("YlOrBr")(9))
plot_scheme(colour("discrete rainbow")(14))

## Plot diagnostic maps
plot_map(colour("bright")(7))
plot_map(colour("sunset")(11))
```

```
plot_map(colour("YlOrBr")(9))
plot_map(colour("discrete rainbow")(14))

## Plot diagnostic images
plot_tiles(colour("discrete rainbow")(14), n = 256)
plot_tiles(colour("discrete rainbow")(23), n = 256)
plot_tiles(colour("smooth rainbow")(256), n = 256)
```

plot_scheme

Plot Color Scheme

Description

Shows colors in a plot.

Usage

```
plot_scheme(x, colours = FALSE, names = FALSE, size = 1)
```

Arguments

x	A character vector of colors.
colours	A logical scalar: should the hexadecimal representation of the colors be displayed?
names	A logical scalar: should the name of the colors be displayed?
size	A numeric value giving the amount by which plotting text should be magnified relative to the default. Works the same as cex parameter of graphics::par() .

Value

plot_scheme() is called for its side-effects: it results in a graphic being displayed (invisibly returns x).

Author(s)

N. Frerebeau

See Also

Other diagnostic tools: [change\(\)](#), [compare\(\)](#), [plot.color_scheme\(\)](#), [plot_map\(\)](#), [plot_scheme_colourblind\(\)](#), [plot_tiles\(\)](#)

Examples

```
plot(colour("bright")(7))
plot(colour("smooth rainbow")(256))

## Plot colour schemes
plot_scheme(colour("bright")(7))
plot_scheme(colour("sunset")(11))
plot_scheme(colour("YlOrBr")(9))
plot_scheme(colour("discrete rainbow")(14))

## Plot diagnostic maps
plot_map(colour("bright")(7))
plot_map(colour("sunset")(11))
plot_map(colour("YlOrBr")(9))
plot_map(colour("discrete rainbow")(14))

## Plot diagnostic images
plot_tiles(colour("discrete rainbow")(14), n = 256)
plot_tiles(colour("discrete rainbow")(23), n = 256)
plot_tiles(colour("smooth rainbow")(256), n = 256)
```

plot_scheme_colourblind

Plot Simulated Color Blindness

Description

Shows colors in a plot with different types of simulated color blindness.

Usage

```
plot_scheme_colourblind(x)
```

```
plot_scheme_colorblind(x)
```

Arguments

x A [character](#) vector of colors.

Value

plot_scheme_colourblind() is called for its side-effects: it results in a graphic being displayed (invisibly returns x).

Author(s)

N. Frerebeau, V. Arel-Bundock

See Also

Other diagnostic tools: [change\(\)](#), [compare\(\)](#), [plot.color_scheme\(\)](#), [plot_map\(\)](#), [plot_scheme\(\)](#), [plot_tiles\(\)](#)

Examples

```
if (requireNamespace("spacesXYZ", quietly = TRUE)) {
  ## Trichromat
  pal <- colour("bright")
  plot_scheme(pal(7))

  ## Deuteranopia
  deu <- change(pal, mode = "deuteranopia")
  plot_scheme(deu(7))

  ## Protanopia
  pro <- change(pal, mode = "protanopia")
  plot_scheme(pro(7))

  ## Tritanopia
  tri <- change(pal, mode = "tritanopia")
  plot_scheme(tri(7))

  ## Achromatopsia
  ach <- change(pal, mode = "achromatopsia")
  plot_scheme(ach(7))

  ## Plot simulated color blindness
  plot_scheme_colorblind(pal(7))
}
```

plot_tiles

Diagnostic Map

Description

Produces a diagnostic map for a given color scheme.

Usage

```
plot_tiles(x, n = 512)
```

Arguments

x A [character](#) vector of colors.

n An [integer](#) specifying the size of the grid (defaults to 512).

Value

plot_tiles() is called for its side-effects: it results in a graphic being displayed (invisibly returns x).

Author(s)

N. Frerebeau

See Also

Other diagnostic tools: [change\(\)](#), [compare\(\)](#), [plot.color_scheme\(\)](#), [plot_map\(\)](#), [plot_scheme\(\)](#), [plot_scheme_colourblind\(\)](#)

Examples

```
plot(colour("bright")(7))
plot(colour("smooth rainbow")(256))

## Plot colour schemes
plot_scheme(colour("bright")(7))
plot_scheme(colour("sunset")(11))
plot_scheme(colour("YlOrBr")(9))
plot_scheme(colour("discrete rainbow")(14))

## Plot diagnostic maps
plot_map(colour("bright")(7))
plot_map(colour("sunset")(11))
plot_map(colour("YlOrBr")(9))
plot_map(colour("discrete rainbow")(14))

## Plot diagnostic images
plot_tiles(colour("discrete rainbow")(14), n = 256)
plot_tiles(colour("discrete rainbow")(23), n = 256)
plot_tiles(colour("smooth rainbow")(256), n = 256)
```

scale_colour_land *AVHRR Global Land Cover Classification Color Scheme for **ggplot2** and **ggraph***

Description

Provides the AVHRR Global Land Cover classification as modified by Paul Tol (colorblind safe).

Usage

```
scale_colour_land(..., lang = "en", aesthetics = "colour")

scale_color_land(..., lang = "en", aesthetics = "colour")
```

```
scale_fill_land(..., lang = "en", aesthetics = "fill")
```

```
scale_edge_colour_land(..., lang = "en")
```

```
scale_edge_color_land(..., lang = "en")
```

```
scale_edge_fill_land(..., lang = "en")
```

Arguments

...	Arguments passed on to <code>ggplot2::discrete_scale()</code> .
lang	A character string specifying the language for the color names (see details). It must be one of "en" (english, the default), "fr" (french) or NULL. If not NULL, the values will be matched based on the color names.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Details

Values will be matched based on the land classification names.

Value

A **discrete** scale.

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other themed color schemes: [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#)

Other qualitative color schemes: [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Examples

```
library(ggplot2)
```

```
land <- data.frame(  
  name = c(  
    "water", "evergreen needleleaf forest", "deciduous needleleaf forest",  
    "mixed forest", "evergreen broadleaf forest", "deciduous broadleaf forest",  
    "woodland", "wooded grassland", "grassland", "cropland", "closed shrubland",
```

```

    "open shrubland", "bare ground", "urban and built"
  )
)

ggplot2::ggplot(land) +
  ggplot2::geom_rect(ggplot2::aes(xmin = rep(0, 14), xmax = rep(1, 14),
                                ymin = 1:14, ymax = 1:14+1, fill = name)) +
  ggplot2::scale_y_reverse() +
  scale_fill_land(name = "land")

```

scale_colour_soil *FAO Soil Reference Groups Color Scheme for **ggplot2** and **ggraph***

Description

Provides the FAO Soil Reference Groups color scheme.

Usage

```
scale_colour_soil(..., lang = "en", aesthetics = "colour")
```

```
scale_color_soil(..., lang = "en", aesthetics = "colour")
```

```
scale_fill_soil(..., lang = "en", aesthetics = "fill")
```

```
scale_edge_colour_soil(..., lang = "en")
```

```
scale_edge_color_soil(..., lang = "en")
```

```
scale_edge_fill_soil(..., lang = "en")
```

Arguments

...	Arguments passed on to <code>ggplot2::discrete_scale()</code> .
lang	A character string specifying the language for the color names (see details). It must be one of "en" (english, the default), "fr" (french) or NULL. If not NULL, the values will be matched based on the color names.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Details

Values will be matched based on the soil names.

Value

A [discrete](#) scale.

Author(s)

N. Frerebeau

References

Jones, A., Montanarella, L. & Jones, R. (Ed.) (2005). *Soil atlas of Europe*. Luxembourg: European Commission, Office for Official Publications of the European Communities. 128 pp. ISBN: 92-894-8120-X.

See Also

Other themed color schemes: [scale_colour_land\(\)](#), [scale_colour_stratigraphy\(\)](#)

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Examples

```
library(ggplot2)

soil <- data.frame(
  name = c(
    "Acrisol", "Albeluvisol", "Andosol", "Anthrosol", "Arenosol", "Calcisol",
    "Cambisol", "Chernozem", "Cryosol", "Fluvisol", "Kastanozem", "Gleysol",
    "Gypsisol", "Histosol", "Leptosol", "Luvisol", "Phaeozem", "Planosol",
    "Podzol", "Regosol", "Solonchak", "Solonetz", "Umbrisol", "Vertisol"
  )
)

ggplot2::ggplot(soil) +
  ggplot2::geom_rect(ggplot2::aes(xmin = rep(0, 24), xmax = rep(1, 24),
                                ymin = 1:24, ymax = 1:24+1, fill = name)) +
  ggplot2::scale_y_reverse() +
  scale_fill_soil(name = "Soil")
```

scale_colour_stratigraphy

Geologic Timescale Color Scheme for ggplot2 and ggraph

Description

Provides the geologic timescale color scheme.

Usage

```
scale_colour_stratigraphy(..., lang = "en", aesthetics = "colour")
scale_color_stratigraphy(..., lang = "en", aesthetics = "colour")
scale_fill_stratigraphy(..., lang = "en", aesthetics = "fill")
scale_edge_colour_stratigraphy(..., lang = "en")
scale_edge_color_stratigraphy(..., lang = "en")
scale_edge_fill_stratigraphy(..., lang = "en")
```

Arguments

...	Arguments passed on to <code>ggplot2::discrete_scale()</code> .
lang	A character string specifying the language for the color names (see details). It must be one of "en" (english, the default), "fr" (french) or NULL. If not NULL, the values will be matched based on the color names.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Details

Values will be matched based on the geological unit names.

Value

A [discrete](#) scale.

Author(s)

N. Frerebeau

References

[Commission for the Geological Map of the World.](#)

See Also

Other themed color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#)

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_okabeito_discrete](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Examples

```

library(ggplot2)

strati <- data.frame(
  name = c("Phanerozoic", "Paleozoic", "Cambrian", "Ordovician", "Silurian",
           "Devonian", "Carboniferous", "Mesozoic", "Triassic", "Cretaceous",
           "Jurassic", "Cenozoic", "Paleogene", "Neogene", "Quaternary"),
  type = c("Eon", "Era", "Period", "Period", "Period", "Period", "Period",
           "Era", "Period", "Period", "Period", "Era", "Period", "Period",
           "Period"),
  start = c(541, 541, 541, 485, 444, 419, 359,
            252, 252, 201, 145, 66, 66, 23, 2.6),
  end = c(0, 252, 485, 444, 419, 359, 252,
          66, 201, 145, 66, 2.6, 23, 2.6, 0)
)

## Keep chronological order in the legend
strati$name <- factor(strati$name, levels = rev(unique(strati$name)),
                    ordered = TRUE)

## Workaround: use `limits = force` to remove unused values
ggplot2::ggplot(strati) +
  ggplot2::geom_rect(ggplot2::aes(xmin = rep(0, 15), xmax = rep(1, 15),
                                  ymin = start, ymax = end, fill = name)) +
  ggplot2::scale_y_reverse() +
  ggplot2::facet_grid(. ~ type) +
  scale_fill_stratigraphy(name = "Stratigraphy", limits = force)

```

scale_crameri_acton *Fabio Crameri's acton Sequential Color Scheme*

Description

Fabio Crameri's *acton* Sequential Color Scheme

Usage

```

scale_colour_acton(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_acton(
  ...,
  reverse = FALSE,
  range = c(0, 1),

```

```

    discrete = FALSE,
    aesthetics = "colour"
  )

scale_fill_acton(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_acton(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_acton(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_acton(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [continuous](#) scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_bam

Fabio Crameri's bam Diverging Color Scheme

Description

Fabio Crameri's *bam* Diverging Color Scheme

Usage

```
scale_colour_bam(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
```

```
    discrete = FALSE,  
    aesthetics = "colour"  
  )  
  
  scale_color_bam(  
    ...,  
    reverse = FALSE,  
    range = c(0, 1),  
    midpoint = 0,  
    discrete = FALSE,  
    aesthetics = "colour"  
  )  
  
  scale_fill_bam(  
    ...,  
    reverse = FALSE,  
    range = c(0, 1),  
    midpoint = 0,  
    discrete = FALSE,  
    aesthetics = "fill"  
  )  
  
  scale_edge_colour_bam(  
    ...,  
    reverse = FALSE,  
    range = c(0, 1),  
    midpoint = 0,  
    discrete = FALSE,  
    aesthetics = "edge_colour"  
  )  
  
  scale_edge_color_bam(  
    ...,  
    reverse = FALSE,  
    range = c(0, 1),  
    midpoint = 0,  
    discrete = FALSE,  
    aesthetics = "edge_colour"  
  )  
  
  scale_edge_fill_bam(  
    ...,  
    reverse = FALSE,  
    range = c(0, 1),  
    midpoint = 0,  
    discrete = FALSE,  
    aesthetics = "edge_fill"  
  )
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_bamako *Fabio Crameri's bamako Sequential Color Scheme*

Description

Fabio Crameri's *bamako* Sequential Color Scheme

Usage

```
scale_colour_bamako(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
```

```
  aesthetics = "colour"
)

scale_color_bamako(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_bamako(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_bamako(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_bamako(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_bamako(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color do-

	main to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [continuous](#) scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bamO](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_brocO](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_corkO](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_romaO](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#), [scale_crameri_vikO](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_bamO *Fabio Crameri's bamO Cyclic Color Scheme*

Description

Fabio Crameri's *bamO* Cyclic Color Scheme

Usage

```
scale_colour_bamO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_color_bamO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_fill_bamO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256

vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other cyclic color schemes: [scale_crameri_broc0](#), [scale_crameri_cork0](#), [scale_crameri_roma0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_batlow *Fabio Crameri's batlow Sequential Color Scheme*

Description

Fabio Crameri's *batlow* Sequential Color Scheme

Usage

```
scale_colour_batlow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_batlow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_batlow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_batlow(
  ...,
  reverse = FALSE,
```

```

    range = c(0, 1),
    discrete = FALSE,
    aesthetics = "edge_colour"
  )

scale_edge_color_batlow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_batlow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256

bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#),

[scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#),
[scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_batlowK *Fabio Crameri's batlowK Sequential Color Scheme*

Description

Fabio Crameri's *batlowK* Sequential Color Scheme

Usage

```
scale_colour_batlowK(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_batlowK(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_batlowK(
  ...,
  reverse = FALSE,
  range = c(0, 1),
```

```

    discrete = FALSE,
    aesthetics = "fill"
  )

scale_edge_colour_batlowK(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_batlowK(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_batlowK(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: `scale_crameri_acton`, `scale_crameri_bam`, `scale_crameri_bam0`, `scale_crameri_bamako`, `scale_crameri_batlow`, `scale_crameri_batlowW`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc`, `scale_crameri_broc0`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork`, `scale_crameri_cork0`, `scale_crameri_davos`, `scale_crameri_devon`, `scale_crameri_fes`, `scale_crameri_grayC`, `scale_crameri_hawaii`, `scale_crameri_imola`, `scale_crameri_lajolla`, `scale_crameri_lapaz`, `scale_crameri_lisbon`, `scale_crameri_nuuk`, `scale_crameri_oleron`, `scale_crameri_oslo`, `scale_crameri_roma`, `scale_crameri_roma0`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimu`, `scale_crameri_vik`, `scale_crameri_vik0`

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_batlowW *Fabio Crameri's batlowW Sequential Color Scheme*

Description

Fabio Crameri's *batlowW* Sequential Color Scheme

Usage

```
scale_colour_batlowW(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_batlowW(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
```

```

  aesthetics = "colour"
)

scale_fill_batlowW(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_batlowW(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_batlowW(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_batlowW(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: `scale_crameri_acton`, `scale_crameri_bamako`, `scale_crameri_batlow`, `scale_crameri_batlowK`, `scale_crameri_bilbao`, `scale_crameri_buda`, `scale_crameri_davos`, `scale_crameri_devon`, `scale_crameri_grayC`, `scale_crameri_hawaii`, `scale_crameri_imola`, `scale_crameri_lajolla`, `scale_crameri_lapaz`, `scale_crameri_nuuk`, `scale_crameri_oslo`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_tol_YlOrBr`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_smoothrainbow`

Other Fabio Crameri's color schemes: `scale_crameri_acton`, `scale_crameri_bam`, `scale_crameri_bam0`, `scale_crameri_bamako`, `scale_crameri_batlow`, `scale_crameri_batlowK`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc`, `scale_crameri_broc0`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork`, `scale_crameri_cork0`, `scale_crameri_davos`, `scale_crameri_devon`, `scale_crameri_fes`, `scale_crameri_grayC`, `scale_crameri_hawaii`, `scale_crameri_imola`, `scale_crameri_lajolla`, `scale_crameri_lapaz`, `scale_crameri_lisbon`, `scale_crameri_nuuk`, `scale_crameri_oleron`, `scale_crameri_oslo`, `scale_crameri_roma`, `scale_crameri_roma0`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimo`, `scale_crameri_vik`, `scale_crameri_vik0`

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_berlin *Fabio Crameri's berlin Diverging Color Scheme*

Description

Fabio Crameri's *berlin* Diverging Color Scheme

Usage

```
scale_colour_berlin(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
```



```
    discrete = FALSE,
    aesthetics = "colour"
  )

scale_color_berlin(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_berlin(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_berlin(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_berlin(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_berlin(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_bilbao *Fabio Crameri's bilbao Sequential Color Scheme*

Description

Fabio Crameri's *bilbao* Sequential Color Scheme

Usage

```
scale_colour_bilbao(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
```

```
  aesthetics = "colour"
)

scale_color_bilbao(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_bilbao(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_bilbao(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_bilbao(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_bilbao(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color do-

	main to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [continuous](#) scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_broc *Fabio Crameri's broc Diverging Color Scheme*

Description

Fabio Crameri's *broc* Diverging Color Scheme

Usage

```
scale_colour_broc(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_broc(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_broc(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_broc(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_broc(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_fill_broc(  
  ...,
```

```

reverse = FALSE,
range = c(0, 1),
midpoint = 0,
discrete = FALSE,
aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

SourceCrameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293**References**Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607**See Also**

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bamO](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_brocO](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_corkO](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_romaO](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vikO](#)

Examples

```
data(economics, package = "ggplot2")
```

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)
```

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_brocO *Fabio Crameri's brocO Cyclic Color Scheme*

Description

Fabio Crameri's *brocO* Cyclic Color Scheme

Usage

```

scale_colour_brocO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_brocO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_brocO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256

vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other cyclic color schemes: [scale_crameri_bam0](#), [scale_crameri_cork0](#), [scale_crameri_roma0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_buda *Fabio Crameri's buda Sequential Color Scheme*

Description

Fabio Crameri's *buda* Sequential Color Scheme

Usage

```
scale_colour_buda(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_buda(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_buda(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_buda(
  ...,
  reverse = FALSE,
```

```

    range = c(0, 1),
    discrete = FALSE,
    aesthetics = "edge_colour"
  )

scale_edge_color_buda(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_buda(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256

bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#),

```
scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku, scale_crameri_vanimo,  
scale_crameri_vik, scale_crameri_vik0
```

Examples

```
data(faithfuld, package = "ggplot2")  
  
ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +  
  ggplot2::geom_raster() +  
  scale_fill_batlow()  
  
ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +  
  ggplot2::geom_raster() +  
  scale_fill_bamako()  
  
ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +  
  ggplot2::geom_raster() +  
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_bukavu *Fabio Crameri's bukavu Multi-Sequential Color Scheme*

Description

Fabio Crameri's *bukavu* Multi-Sequential Color Scheme

Usage

```
scale_colour_bukavu(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  aesthetics = "colour"  
)  
  
scale_color_bukavu(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  aesthetics = "colour"  
)  
  
scale_fill_bukavu(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),
```

```
midpoint = 0,
aesthetics = "fill"
)
```

Arguments

... Arguments passed to `ggplot2::continuous_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

range A length-two **numeric** vector specifying the fraction of the scheme's color domain to keep.

midpoint A length-one **numeric** vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other multi sequential color schemes: [scale_crameri_fes](#), [scale_crameri_oleron](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(volcano)

volcan <- data.frame(
  x = rep(1:ncol(volcano), each = nrow(volcano)),
  y = rep(1:nrow(volcano), times = ncol(volcano)),
  z = as.numeric(volcano)
)

ggplot2::ggplot(volcan, ggplot2::aes(x, y, fill = z)) +
  ggplot2::geom_raster() +
  scale_fill_oleron(midpoint = 125)
```

scale_crameri_cork *Fabio Crameri's cork Diverging Color Scheme*

Description

Fabio Crameri's *cork* Diverging Color Scheme

Usage

```
scale_colour_cork(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_cork(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_cork(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_cork(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_cork(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_fill_cork(  
  ...,
```

```

reverse = FALSE,
range = c(0, 1),
midpoint = 0,
discrete = FALSE,
aesthetics = "edge_fill"
)

```

Arguments

... Arguments passed to `ggplot2::continuous_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

range A length-two **numeric** vector specifying the fraction of the scheme's color domain to keep.

midpoint A length-one **numeric** vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.

discrete A **logical** scalar: should the color scheme be used as a discrete scale?

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")
```

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)
```

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_cork0 *Fabio Crameri's corkO Cyclic Color Scheme*

Description

Fabio Crameri's *corkO* Cyclic Color Scheme

Usage

```
scale_colour_corkO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_color_corkO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_fill_corkO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256

vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other cyclic color schemes: [scale_crameri_bam0](#), [scale_crameri_broc0](#), [scale_crameri_roma0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_davos *Fabio Crameri's davos Sequential Color Scheme*

Description

Fabio Crameri's *davos* Sequential Color Scheme

Usage

```
scale_colour_davos(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_davos(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_davos(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_davos(
  ...,
  reverse = FALSE,
```

```

    range = c(0, 1),
    discrete = FALSE,
    aesthetics = "edge_colour"
  )

scale_edge_color_davos(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_davos(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256

bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#),

[scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#),
[scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_devon *Fabio Crameri's devon Sequential Color Scheme*

Description

Fabio Crameri's *devon* Sequential Color Scheme

Usage

```
scale_colour_devon(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_devon(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_devon(
  ...,
  reverse = FALSE,
  range = c(0, 1),
```

```

    discrete = FALSE,
    aesthetics = "fill"
  )

  scale_edge_colour_devon(
    ...,
    reverse = FALSE,
    range = c(0, 1),
    discrete = FALSE,
    aesthetics = "edge_colour"
  )

  scale_edge_color_devon(
    ...,
    reverse = FALSE,
    range = c(0, 1),
    discrete = FALSE,
    aesthetics = "edge_colour"
  )

  scale_edge_fill_devon(
    ...,
    reverse = FALSE,
    range = c(0, 1),
    discrete = FALSE,
    aesthetics = "edge_fill"
  )

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: `scale_crameri_acton`, `scale_crameri_bam`, `scale_crameri_bam0`, `scale_crameri_bamako`, `scale_crameri_batlow`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc`, `scale_crameri_broc0`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork`, `scale_crameri_cork0`, `scale_crameri_davos`, `scale_crameri_fes`, `scale_crameri_grayC`, `scale_crameri_hawaii`, `scale_crameri_imola`, `scale_crameri_lajolla`, `scale_crameri_lapaz`, `scale_crameri_lisbon`, `scale_crameri_nuuk`, `scale_crameri_oleron`, `scale_crameri_oslo`, `scale_crameri_roma`, `scale_crameri_roma0`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimo`, `scale_crameri_vik`, `scale_crameri_vik0`

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_fes *Fabio Crameri's fes Multi-Sequential Color Scheme*

Description

Fabio Crameri's *fes* Multi-Sequential Color Scheme

Usage

```
scale_colour_fes(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  aesthetics = "colour"
)

scale_color_fes(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
```

```

  aesthetics = "colour"
)

scale_fill_fes(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  aesthetics = "fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256

turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other multi sequential color schemes: [scale_crameri_bukavu](#), [scale_crameri_oleron](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(volcano)

volcan <- data.frame(
  x = rep(1:ncol(volcano), each = nrow(volcano)),
  y = rep(1:nrow(volcano), times = ncol(volcano)),
  z = as.numeric(volcano)
)

ggplot2::ggplot(volcan, ggplot2::aes(x, y, fill = z)) +
  ggplot2::geom_raster() +
  scale_fill_oleron(midpoint = 125)
```

scale_crameri_grayC *Fabio Crameri's grayC Sequential Color Scheme*

Description

Fabio Crameri's *grayC* Sequential Color Scheme

Usage

```
scale_colour_grayC(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_grayC(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_grayC(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_grayC(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_grayC(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"
```



```

)

scale_edge_fill_grayC(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256

oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")
```

```
ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()
```

```
ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
```

```
ggplot2::geom_raster() +  
  scale_fill_bamako()  
  
ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +  
  ggplot2::geom_raster() +  
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_hawaii *Fabio Crameri's hawaii Sequential Color Scheme*

Description

Fabio Crameri's *hawaii* Sequential Color Scheme

Usage

```
scale_colour_hawaii(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_hawaii(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_hawaii(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_hawaii(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)
```

```

scale_edge_color_hawaii(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_hawaii(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256

hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_imola *Fabio Crameri's imola Sequential Color Scheme*

Description

Fabio Crameri's *imola* Sequential Color Scheme

Usage

```
scale_colour_imola(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_imola(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_imola(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)
```

```

scale_edge_colour_imola(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_imola(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_imola(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256

devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#),


```
scale_crameri_buda, scale_crameri_bukavu, scale_crameri_cork, scale_crameri_cork0,
scale_crameri_davos, scale_crameri_devon, scale_crameri_fes, scale_crameri_grayC, scale_crameri_hawaii,
scale_crameri_lajolla, scale_crameri_lapaz, scale_crameri_lisbon, scale_crameri_nuuk,
scale_crameri_oleron, scale_crameri_oslo, scale_crameri_roma, scale_crameri_roma0,
scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku, scale_crameri_vanimo,
scale_crameri_vik, scale_crameri_vik0
```

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_lajolla *Fabio Crameri's lajolla Sequential Color Scheme*

Description

Fabio Crameri's *lajolla* Sequential Color Scheme

Usage

```
scale_colour_lajolla(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_lajolla(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)
```

```

scale_fill_lajolla(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_lajolla(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_lajolla(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_lajolla(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: `scale_crameri_acton`, `scale_crameri_bamako`, `scale_crameri_batlow`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_bilbao`, `scale_crameri_buda`, `scale_crameri_davos`, `scale_crameri_devon`, `scale_crameri_grayC`, `scale_crameri_hawaii`, `scale_crameri_imola`, `scale_crameri_lapaz`, `scale_crameri_nuuk`, `scale_crameri_oslo`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_tol_Yl0rBr`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_smoothrainbow`

Other Fabio Crameri's color schemes: `scale_crameri_acton`, `scale_crameri_bam`, `scale_crameri_bam0`, `scale_crameri_bamako`, `scale_crameri_batlow`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc`, `scale_crameri_broc0`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork`, `scale_crameri_cork0`, `scale_crameri_davos`, `scale_crameri_devon`, `scale_crameri_fes`, `scale_crameri_grayC`, `scale_crameri_hawaii`, `scale_crameri_imola`, `scale_crameri_lapaz`, `scale_crameri_lisbon`, `scale_crameri_nuuk`, `scale_crameri_oleron`, `scale_crameri_oslo`, `scale_crameri_roma`, `scale_crameri_roma0`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanim0`, `scale_crameri_vik`, `scale_crameri_vik0`

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_lapaz *Fabio Crameri's lapaz Sequential Color Scheme*

Description

Fabio Crameri's *lapaz* Sequential Color Scheme

Usage

```
scale_colour_lapaz(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
```

```
  aesthetics = "colour"
)

scale_color_lapaz(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_lapaz(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_lapaz(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_lapaz(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_lapaz(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color do-

	main to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [continuous](#) scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_lisbon *Fabio Crameri's lisbon Diverging Color Scheme*

Description

Fabio Crameri's *lisbon* Diverging Color Scheme

Usage

```
scale_colour_lisbon(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_lisbon(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_lisbon(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_lisbon(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_lisbon(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_fill_lisbon(  
  ...,
```



```

reverse = FALSE,
range = c(0, 1),
midpoint = 0,
discrete = FALSE,
aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")
```

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)
```

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_nuuk

Fabio Crameri's nuuk Sequential Color Scheme

Description

Fabio Crameri's *nuuk* Sequential Color Scheme

Usage

```
scale_colour_nuuk(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_nuuk(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_nuuk(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_nuuk(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_nuuk(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_fill_nuuk(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_fill"  
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_oleron *Fabio Crameri's oleron Multi-Sequential Color Scheme*

Description

Fabio Crameri's *oleron* Multi-Sequential Color Scheme

Usage

```
scale_colour_oleron(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  aesthetics = "colour"
)
```

```
scale_color_oleron(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  aesthetics = "colour"
)
```

```
scale_fill_oleron(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  aesthetics = "fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other multi sequential color schemes: [scale_crameri_bukavu](#), [scale_crameri_fes](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(volcano)

volcan <- data.frame(
  x = rep(1:ncol(volcano), each = nrow(volcano)),
  y = rep(1:nrow(volcano), times = ncol(volcano)),
  z = as.numeric(volcano)
)

ggplot2::ggplot(volcan, ggplot2::aes(x, y, fill = z)) +
  ggplot2::geom_raster() +
  scale_fill_oleron(midpoint = 125)
```

scale_crameri_oslo *Fabio Crameri's oslo Sequential Color Scheme*

Description

Fabio Crameri's *oslo* Sequential Color Scheme

Usage

```
scale_colour_oslo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_oslo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
```



```

    discrete = FALSE,
    aesthetics = "colour"
  )

scale_fill_oslo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_oslo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_oslo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_oslo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [continuous](#) scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. [doi:10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. [doi:10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanim0](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_roma *Fabio Crameri's roma Diverging Color Scheme*

Description

Fabio Crameri's *roma* Diverging Color Scheme

Usage

```
scale_colour_roma(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
```

```
    discrete = FALSE,
    aesthetics = "colour"
  )

scale_color_roma(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_roma(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_roma(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_roma(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_roma(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)
```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bamO](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_romaO *Fabio Crameri's romaO Cyclic Color Scheme*

Description

Fabio Crameri's *romaO* Cyclic Color Scheme

Usage

```
scale_colour_romaO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
```

```

  aesthetics = "colour"
)

scale_color_roma0(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_roma0(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256

broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other cyclic color schemes: [scale_crameri_bam0](#), [scale_crameri_broc0](#), [scale_crameri_cork0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_tofino *Fabio Crameri's tofino Diverging Color Scheme*

Description

Fabio Crameri's *tofino* Diverging Color Scheme

Usage

```
scale_colour_tofino(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_tofino(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_tofino(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_tofino(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_tofino(  
  ...,
```

```

reverse = FALSE,
range = c(0, 1),
midpoint = 0,
discrete = FALSE,
aesthetics = "edge_colour"
)

scale_edge_fill_tofino(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256

broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:[10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:[10.5194/gmd1125412018](https://doi.org/10.5194/gmd1125412018)

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:[10.1038/s41467020191607](https://doi.org/10.1038/s41467020191607)

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_tokyo *Fabio Crameri's tokyo Sequential Color Scheme*

Description

Fabio Crameri's *tokyo* Sequential Color Scheme

Usage

```
scale_colour_tokyo(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_tokyo(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_tokyo(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_tokyo(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_tokyo(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"
```

```

)

scale_edge_fill_tokyo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256
hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256

oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_turku](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
```

```
ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_turku *Fabio Crameri's turku Sequential Color Scheme*

Description

Fabio Crameri's *turku* Sequential Color Scheme

Usage

```
scale_colour_turku(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_turku(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_turku(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_turku(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)
```

```

scale_edge_color_turku(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_turku(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
batlow	256
batlowW	256
batlowK	256
devon	256
lajolla	256
bamako	256
davos	256
bilbao	256
nuuk	256
oslo	256
grayC	256

hawaii	256
lapaz	256
tokyo	256
buda	256
acton	256
turku	256
imola	256
oleron*	256
bukavu*	256
fes*	256

*: multisequential color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_crameri_vik0](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_batlow()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_bamako()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_hawaii(reverse = TRUE)
```

scale_crameri_vanimo *Fabio Crameri's vanimo Diverging Color Scheme*

Description

Fabio Crameri's *vanimo* Diverging Color Scheme

Usage

```
scale_colour_vanimo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_color_vanimo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_fill_vanimo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
```

```

    discrete = FALSE,
    aesthetics = "fill"
  )

scale_edge_colour_vanimo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_vanimo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_vanimo(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#),

```
scale_crameri_imola, scale_crameri_lajolla, scale_crameri_lapaz, scale_crameri_lisbon,
scale_crameri_nuuk, scale_crameri_oleron, scale_crameri_oslo, scale_crameri_roma, scale_crameri_roma0,
scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku, scale_crameri_vik,
scale_crameri_vik0
```

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_vik *Fabio Crameri's vik Diverging Color Scheme*

Description

Fabio Crameri's *vik* Diverging Color Scheme

Usage

```
scale_colour_vik(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_color_vik(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)
```

```
scale_fill_vik(
  ...,
  reverse = FALSE,
  range = c(0, 1),
```

```

midpoint = 0,
discrete = FALSE,
aesthetics = "fill"
)

scale_edge_colour_vik(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_vik(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_vik(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. doi:10.5281/zenodo.4491293

References

Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018

Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#),

```
scale_crameri_imola, scale_crameri_lajolla, scale_crameri_lapaz, scale_crameri_lisbon,
scale_crameri_nuuk, scale_crameri_oleron, scale_crameri_oslo, scale_crameri_roma, scale_crameri_roma0,
scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku, scale_crameri_vanimu,
scale_crameri_vikO
```

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_crameri_vikO *Fabio Crameri's vikO Cyclic Color Scheme*

Description

Fabio Crameri's *vikO* Cyclic Color Scheme

Usage

```
scale_colour_vikO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_vikO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_vikO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)
```


Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.
broc	256
cork	256
vik	256
lisbon	256
tofino	256
berlin	256
roma	256
bam	256
vanimo	256
broc0*	256
cork0*	256
vik0*	256
roma0*	256
bam0*	256

*: cyclic color schemes.

Author(s)

N. Frerebeau

Source

Crameri, F. (2021). Scientific colour maps. *Zenodo*, v7.0. [doi:10.5281/zenodo.4491293](https://doi.org/10.5281/zenodo.4491293)

References

- Crameri, F. (2018). Geodynamic diagnostics, scientific visualisation and StagLab 3.0. *Geosci. Model Dev.*, 11, 2541-2562. doi:10.5194/gmd1125412018
- Crameri, F., Shephard, G. E. & Heron, P. J. (2020). The misuse of colour in science communication. *Nature Communications*, 11, 5444. doi:10.1038/s41467020191607

See Also

Other cyclic color schemes: [scale_crameri_bam0](#), [scale_crameri_broc0](#), [scale_crameri_cork0](#), [scale_crameri_roma0](#)

Other Fabio Crameri's color schemes: [scale_crameri_acton](#), [scale_crameri_bam](#), [scale_crameri_bam0](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_berlin](#), [scale_crameri_bilbao](#), [scale_crameri_broc](#), [scale_crameri_broc0](#), [scale_crameri_buda](#), [scale_crameri_bukavu](#), [scale_crameri_cork](#), [scale_crameri_cork0](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_fes](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_lisbon](#), [scale_crameri_nuuk](#), [scale_crameri_oleron](#), [scale_crameri_oslo](#), [scale_crameri_roma](#), [scale_crameri_roma0](#), [scale_crameri_tofino](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_crameri_vanimu](#), [scale_crameri_vik](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_colour_berlin(midpoint = 9000)
```

scale_okabeito_discrete

*Okabe and Ito's Discrete Color Scheme for **ggplot2** and **ggraph***

Description

Provides the qualitative color scale from Okabe and Ito 2008.

Usage

```
scale_colour_okabeito(
  ...,
  reverse = FALSE,
  black_position = c("first", "last"),
  aesthetics = "colour"
```

```

)

scale_color_okabeito(
  ...,
  reverse = FALSE,
  black_position = c("first", "last"),
  aesthetics = "colour"
)

scale_fill_okabeito(
  ...,
  reverse = FALSE,
  black_position = c("first", "last"),
  aesthetics = "fill"
)

scale_edge_colour_okabeito(
  ...,
  reverse = FALSE,
  black_position = c("first", "last"),
  aesthetics = "edge_colour"
)

scale_edge_color_okabeito(
  ...,
  reverse = FALSE,
  black_position = c("first", "last"),
  aesthetics = "edge_colour"
)

scale_edge_fill_okabeito(
  ...,
  reverse = FALSE,
  black_position = c("first", "last"),
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::discrete_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
black_position	A character string giving the position of the black color. It must be one of "first" or "last". Any unambiguous substring can be given.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Details

This qualitative color scheme is used as given (no interpolation): colors are picked up to the maximum number of supported values (8).

Value

A [discrete](#) scale.

Author(s)

N. Frerebeau

References

Okabe, M. & Ito, K. (2008). *Color Universal Design (CUD): How to Make Figures and Presentations That Are Friendly to Colorblind People*. URL: <https://jfly.uni-koeln.de/color/>.

See Also

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Examples

```
library(ggplot2)

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_okabeito()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_okabeito(black_position = "last")
```

scale_picker

Color Scale Builder

Description

Builds a color scale for **ggplot2** or **ggraph**.

Usage

```
scale_colour_picker(..., palette = "YlOrBr")
scale_color_picker(..., palette = "YlOrBr")
scale_fill_picker(..., palette = "YlOrBr")
scale_edge_colour_picker(..., palette = "YlOrBr")
scale_edge_color_picker(..., palette = "YlOrBr")
scale_edge_fill_picker(..., palette = "YlOrBr")
```

Arguments

... Extra parameters to be passed to the color scale function.

palette A [character](#) string giving the name of the color scheme to be used (see [info\(\)](#)).

Value

A [discrete](#) or [continuous](#) scale.

Author(s)

N. Frerebeau

Examples

```
library(ggplot2)

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_picker(palette = "okabeito")
```

scale_tol_bright *Paul Tol's bright Discrete Color Scheme*

Description

Paul Tol's *bright* Discrete Color Scheme

Usage

```

scale_colour_bright(..., reverse = FALSE, aesthetics = "colour")

scale_color_bright(..., reverse = FALSE, aesthetics = "colour")

scale_fill_bright(..., reverse = FALSE, aesthetics = "fill")

scale_edge_colour_bright(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_color_bright(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_fill_bright(..., reverse = FALSE, aesthetics = "edge_fill")

```

Arguments

`...` Arguments passed to `ggplot2::discrete_scale()`.

`reverse` A [logical](#) scalar. Should the resulting vector of colors be reversed?

`aesthetics` A [character](#) string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
 URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

 scale_tol_BuRd

Paul Tol's BuRd Diverging Color Scheme

Description

Paul Tol's *BuRd* Diverging Color Scheme

Usage

```
scale_colour_BuRd(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_BuRd(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_BuRd(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_BuRd(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_BuRd(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_fill_BuRd(  
  ...,
```



```

reverse = FALSE,
range = c(0, 1),
midpoint = 0,
discrete = FALSE,
aesthetics = "edge_fill"
)

```

Arguments

... Arguments passed to `ggplot2::continuous_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

range A length-two **numeric** vector specifying the fraction of the scheme's color domain to keep.

midpoint A length-one **numeric** vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.

discrete A **logical** scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
sunset	11	#FFFFFF
nightfall	17	#FFFFFF
BuRd	9	#FFEE99
PRGn	9	#FFEE99

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1. URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Paul Tol's color schemes: [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_sunset(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_BuRd(midpoint = 9000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_PRGn(midpoint = 9000, range = c(0.25, 1))
```

scale_tol_dark

Paul Tol's dark Discrete Color Scheme

Description

Paul Tol's *dark* Discrete Color Scheme

Usage

```
scale_colour_dark(..., reverse = FALSE, aesthetics = "colour")

scale_color_dark(..., reverse = FALSE, aesthetics = "colour")

scale_fill_dark(..., reverse = FALSE, aesthetics = "fill")

scale_edge_colour_dark(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_color_dark(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_fill_dark(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

...	Arguments passed to <code>ggplot2::discrete_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **discrete** scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2. URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's color schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_nightfall`, `scale_tol_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

scale_tol_discreterainbow

Paul Tol's discrete rainbow Sequential Color Scheme

Description

Paul Tol's *discrete rainbow* Sequential Color Scheme

Usage

```
scale_colour_discreterainbow(..., reverse = FALSE, aesthetics = "colour")

scale_color_discreterainbow(..., reverse = FALSE, aesthetics = "colour")

scale_fill_discreterainbow(..., reverse = FALSE, aesthetics = "fill")

scale_edge_colour_discreterainbow(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
```

```

)

scale_edge_color_discreterainbow(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_discreterainbow(..., reverse = FALSE, aesthetics = "edge_fill")

```

Arguments

... Arguments passed to `ggplot2::discrete_scale()`.

reverse A [logical](#) scalar. Should the resulting vector of colors be reversed?

aesthetics A [character](#) string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
YlOrBr	9	#888888
iridescent	23	#999999
discreterainbow	23	#777777
smoothrainbow	34	#666666

Rainbow Color Scheme

As a general rule, ordered data should not be represented using a rainbow scheme. There are three main arguments against such use (Tol 2018):

- The spectral order of visible light carries no inherent magnitude message.
- Some bands of almost constant hue with sharp transitions between them, can be perceived as jumps in the data.
- Color-blind people have difficulty distinguishing some colors of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two color schemes that are reasonably clear in color-blind vision. To remain color-blind safe, these two schemes must comply with the following conditions:

`discreterainbow` This scheme must not be interpolated.

`smoothrainbow` This scheme does not have to be used over the full range.

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
 URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's color schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_nightfall`, `scale_tol_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

scale_tol_highcontrast

Paul Tol's high contrast Discrete Color Scheme

Description

Paul Tol's *high contrast* Discrete Color Scheme

Usage

```

scale_colour_highcontrast(..., reverse = FALSE, aesthetics = "colour")

scale_color_highcontrast(..., reverse = FALSE, aesthetics = "colour")

scale_fill_highcontrast(..., reverse = FALSE, aesthetics = "fill")

scale_edge_colour_highcontrast(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_highcontrast(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_fill_highcontrast(..., reverse = FALSE, aesthetics = "edge_fill")

```

Arguments

... Arguments passed to `ggplot2::discrete_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **discrete** scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's color schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_nightfall`, `scale_tol_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

`scale_tol_incandescent`*Paul Tol's incandescent Sequential Color Scheme*

Description

Paul Tol's *incandescent* Sequential Color Scheme

Usage

```
scale_colour_incandescent(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)
```

```
scale_color_incandescent(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)
```

```
scale_fill_incandescent(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)
```

```
scale_edge_colour_incandescent(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)
```

```
scale_edge_color_incandescent(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,
```

```

  aesthetics = "edge_colour"
)

scale_edge_fill_incandescent(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

... Arguments passed to `ggplot2::continuous_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

range A length-two **numeric** vector specifying the fraction of the scheme's color domain to keep.

discrete A **logical** scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
YlOrBr	9	#888888
iridescent	23	#999999
discreterainbow	23	#777777
smoothrainbow	34	#666666

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1. URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_Yl0rBr](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_Yl0rBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_Yl0rBr()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_iridescent(reverse = TRUE)

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_smoothrainbow(range = c(0.25, 1))
```

scale_tol_iridescent *Paul Tol's iridescent Sequential Color Scheme*

Description

Paul Tol's *iridescent* Sequential Color Scheme

Usage

```
scale_colour_iridescent(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_iridescent(
  ...,
```

```

reverse = FALSE,
range = c(0, 1),
discrete = FALSE,
aesthetics = "colour"
)

scale_fill_iridescent(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_iridescent(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_iridescent(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_iridescent(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
YlOrBr	9	#888888
iridescent	23	#999999
discreterainbow	23	#777777
smoothrainbow	34	#666666

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_YlOrBr](#), [scale_tol_incandescent](#), [scale_tol_smoothrainbow](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_YlOrBr()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
```

```

ggplot2::geom_raster() +
  scale_fill_iridescent(reverse = TRUE)

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_smoothrainbow(range = c(0.25, 1))

```

scale_tol_light

Paul Tol's light Discrete Color Scheme

Description

Paul Tol's *light* Discrete Color Scheme

Usage

```

scale_colour_light(..., reverse = FALSE, aesthetics = "colour")
scale_color_light(..., reverse = FALSE, aesthetics = "colour")
scale_fill_light(..., reverse = FALSE, aesthetics = "fill")
scale_edge_colour_light(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_color_light(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_fill_light(..., reverse = FALSE, aesthetics = "edge_fill")

```

Arguments

... Arguments passed to `ggplot2::discrete_scale()`.

reverse A [logical](#) scalar. Should the resulting vector of colors be reversed?

aesthetics A [character](#) string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3

vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()
```

```
data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

scale_tol_mediumcontrast

Paul Tol's medium contrast Discrete Color Scheme

Description

Paul Tol's *medium contrast* Discrete Color Scheme

Usage

```
scale_colour_mediumcontrast(..., reverse = FALSE, aesthetics = "colour")

scale_color_mediumcontrast(..., reverse = FALSE, aesthetics = "colour")

scale_fill_mediumcontrast(..., reverse = FALSE, aesthetics = "fill")

scale_edge_colour_mediumcontrast(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_mediumcontrast(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_mediumcontrast(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

...	Arguments passed to <code>ggplot2::discrete_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's color schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_light`, `scale_tol_muted`, `scale_tol_nightfall`, `scale_tol_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
data(mpg, package = "ggplot2")
```

```

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()

```

scale_tol_muted

Paul Tol's muted Discrete Color Scheme

Description

Paul Tol's *muted* Discrete Color Scheme

Usage

```

scale_colour_muted(..., reverse = FALSE, aesthetics = "colour")

scale_color_muted(..., reverse = FALSE, aesthetics = "colour")

scale_fill_muted(..., reverse = FALSE, aesthetics = "fill")

scale_edge_colour_muted(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_color_muted(..., reverse = FALSE, aesthetics = "edge_colour")

scale_edge_fill_muted(..., reverse = FALSE, aesthetics = "edge_fill")

```

Arguments

...	Arguments passed to <code>ggplot2::discrete_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_pale](#), [scale_tol_vibrant](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

scale_tol_nightfall *Paul Tol's nightfall Diverging Color Scheme*

Description

Paul Tol's *nightfall* Diverging Color Scheme

Usage

```
scale_colour_nightfall(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_nightfall(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_nightfall(
  ...,
  reverse = FALSE,
  range = c(0, 1),
```

```

    midpoint = 0,
    discrete = FALSE,
    aesthetics = "fill"
  )

scale_edge_colour_nightfall(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_nightfall(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_nightfall(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [continuous](#) scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
sunset	11	#FFFFFF
nightfall	17	#FFFFFF
BuRd	9	#FFEE99
PRGn	9	#FFEE99

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_sunset](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_color_sunset(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_color_BuRd(midpoint = 9000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemployment)) +
  ggplot2::geom_point() +
  scale_color_PRGn(midpoint = 9000, range = c(0.25, 1))
```

scale_tol_pale *Paul Tol's pale Discrete Color Scheme*

Description

Paul Tol's *pale* Discrete Color Scheme

Usage

```
scale_colour_pale(..., reverse = FALSE, aesthetics = "colour")
scale_color_pale(..., reverse = FALSE, aesthetics = "colour")
scale_fill_pale(..., reverse = FALSE, aesthetics = "fill")
scale_edge_colour_pale(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_color_pale(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_fill_pale(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

... Arguments passed to `ggplot2::discrete_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **discrete** scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2. URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_vibrant`

Other Paul Tol's color schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_nightfall`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

scale_tol_PRGn	<i>Paul Tol's PRGn Diverging Color Scheme</i>
----------------	---

Description

Paul Tol's *PRGn* Diverging Color Scheme

Usage

```
scale_colour_PRGn(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_PRGn(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_PRGn(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_PRGn(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_PRGn(  
  ...,
```

```

reverse = FALSE,
range = c(0, 1),
midpoint = 0,
discrete = FALSE,
aesthetics = "edge_colour"
)

scale_edge_fill_PRGn(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

... Arguments passed to `ggplot2::continuous_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

range A length-two **numeric** vector specifying the fraction of the scheme's color domain to keep.

midpoint A length-one **numeric** vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.

discrete A **logical** scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
sunset	11	#FFFFFF
nightfall	17	#FFFFFF
BuRd	9	#FFEE99
PRGn	9	#FFEE99

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
 URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_nightfall](#), [scale_tol_sunset](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_sunset(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_BuRd(midpoint = 9000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_PRGn(midpoint = 9000, range = c(0.25, 1))
```

scale_tol_smoothrainbow

Paul Tol's smooth rainbow Sequential Color Scheme

Description

Paul Tol's *smooth rainbow* Sequential Color Scheme

Usage

```
scale_colour_smoothrainbow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)
```

```

scale_color_smoothrainbow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_smoothrainbow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_smoothrainbow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_smoothrainbow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_smoothrainbow(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale? If TRUE,

it is a departure from Paul Tol's recommendations and likely a very poor use of color.

aesthetics A **character** string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
YlOrBr	9	#888888
iridescent	23	#999999
discreterainbow	23	#777777
smoothrainbow	34	#666666

Rainbow Color Scheme

As a general rule, ordered data should not be represented using a rainbow scheme. There are three main arguments against such use (Tol 2018):

- The spectral order of visible light carries no inherent magnitude message.
- Some bands of almost constant hue with sharp transitions between them, can be perceived as jumps in the data.
- Color-blind people have difficulty distinguishing some colors of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two color schemes that are reasonably clear in color-blind vision. To remain color-blind safe, these two schemes must comply with the following conditions:

discreterainbow This scheme must not be interpolated.

smoothrainbow This scheme does not have to be used over the full range.

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1. URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_Yl0rBr](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_Yl0rBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_Yl0rBr()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_iridescent(reverse = TRUE)

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_smoothrainbow(range = c(0.25, 1))
```

scale_tol_sunset

Paul Tol's sunset Diverging Color Scheme

Description

Paul Tol's *sunset* Diverging Color Scheme

Usage

```
scale_colour_sunset(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_sunset(
```

```
    ...,
    reverse = FALSE,
    range = c(0, 1),
    midpoint = 0,
    discrete = FALSE,
    aesthetics = "colour"
  )

scale_fill_sunset(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "fill"
)

scale_edge_colour_sunset(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_sunset(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_sunset(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  midpoint = 0,
  discrete = FALSE,
  aesthetics = "edge_fill"
)
```

Arguments

... Arguments passed to `ggplot2::continuous_scale()`.

reverse A **logical** scalar. Should the resulting vector of colors be reversed?

range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Diverging Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
sunset	11	#FFFFFF
nightfall	17	#FFFFFF
BuRd	9	#FFEE99
PRGn	9	#FFEE99

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other diverging color schemes: [scale_crameri_bam](#), [scale_crameri_berlin](#), [scale_crameri_broc](#), [scale_crameri_cork](#), [scale_crameri_lisbon](#), [scale_crameri_roma](#), [scale_crameri_tofino](#), [scale_crameri_vanimo](#), [scale_crameri_vik](#), [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_nightfall](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_vibrant](#)

Examples

```

data(economics, package = "ggplot2")

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_sunset(reverse = TRUE, midpoint = 12000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_BuRd(midpoint = 9000)

ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +
  ggplot2::geom_point() +
  scale_color_PRGn(midpoint = 9000, range = c(0.25, 1))

```

scale_tol_vibrant *Paul Tol's vibrant Discrete Color Scheme*

Description

Paul Tol's *vibrant* Discrete Color Scheme

Usage

```

scale_colour_vibrant(..., reverse = FALSE, aesthetics = "colour")
scale_color_vibrant(..., reverse = FALSE, aesthetics = "colour")
scale_fill_vibrant(..., reverse = FALSE, aesthetics = "fill")
scale_edge_colour_vibrant(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_color_vibrant(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_fill_vibrant(..., reverse = FALSE, aesthetics = "edge_fill")

```

Arguments

...	Arguments passed to <code>ggplot2::discrete_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Qualitative Color Schemes

The qualitative color schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Palette	Max.
bright	7
highcontrast	3
vibrant	7
muted	9
mediumcontrast	6
pale	6
dark	6
light	9

According to Paul Tol's technical note, the bright, highcontrast, vibrant and muted color schemes are color-blind safe. The mediumcontrast color scheme is designed for situations needing color pairs.

The light color scheme is reasonably distinct for both normal or colorblind vision and is intended to fill labeled cells.

The pale and dark schemes are not very distinct in either normal or colorblind vision and should be used as a text background or to highlight a cell in a table.

Refer to the original document for details about the recommended uses (see references)

Author(s)

N. Frerebeau

References

Tol, P. (2021). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.2.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other qualitative color schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_pale](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#)

Examples

```
data(mpg, package = "ggplot2")

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_bright()

ggplot2::ggplot(mpg, ggplot2::aes(displ, hwy, colour = class)) +
  ggplot2::geom_point() +
  scale_colour_vibrant()

data(diamonds, package = "ggplot2")

ggplot2::ggplot(diamonds, ggplot2::aes(clarity, fill = cut)) +
  ggplot2::geom_bar() +
  scale_fill_muted()
```

scale_tol_YlOrBr

Paul Tol's YlOrBr Sequential Color Scheme

Description

Paul Tol's *YlOrBr* Sequential Color Scheme

Usage

```
scale_colour_YlOrBr(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_YlOrBr(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_YlOrBr(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)
```

```

)

scale_edge_colour_YlOrBr(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_color_YlOrBr(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_colour"
)

scale_edge_fill_YlOrBr(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "edge_fill"
)

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's color domain to keep.
discrete	A logical scalar: should the color scheme be used as a discrete scale? If TRUE, it is a departure from Paul Tol's recommendations and likely a very poor use of color.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Sequential Color Schemes

If more colors than defined are needed from a given scheme, the color coordinates are linearly interpolated to provide a continuous version of the scheme.

Palette	Max.	NA value
YlOrBr	9	#888888
iridescent	23	#999999
discreterainbow	23	#777777
smoothrainbow	34	#666666

Author(s)

N. Frerebeau

References

Tol, P. (2018). *Colour Schemes*. SRON. Technical Note No. SRON/EPS/TN/09-002, issue 3.1.
URL: <https://personal.sron.nl/~pault/data/colourschemes.pdf>

See Also

Other sequential color schemes: [scale_crameri_acton](#), [scale_crameri_bamako](#), [scale_crameri_batlow](#), [scale_crameri_batlowK](#), [scale_crameri_batlowW](#), [scale_crameri_bilbao](#), [scale_crameri_buda](#), [scale_crameri_davos](#), [scale_crameri_devon](#), [scale_crameri_grayC](#), [scale_crameri_hawaii](#), [scale_crameri_imola](#), [scale_crameri_lajolla](#), [scale_crameri_lapaz](#), [scale_crameri_nuuk](#), [scale_crameri_oslo](#), [scale_crameri_tokyo](#), [scale_crameri_turku](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Paul Tol's color schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [scale_tol_incandescent](#), [scale_tol_iridescent](#), [scale_tol_light](#), [scale_tol_mediumcontrast](#), [scale_tol_muted](#), [scale_tol_nightfall](#), [scale_tol_pale](#), [scale_tol_smoothrainbow](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

```
data(faithfuld, package = "ggplot2")

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_YlOrBr()

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_iridescent(reverse = TRUE)

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_smoothrainbow(range = c(0.25, 1))
```

Index

* Fabio Crameri's color schemes

- scale_crameri_acton, 27
- scale_crameri_bam, 30
- scale_crameri_bamako, 33
- scale_crameri_bam0, 36
- scale_crameri_batlow, 39
- scale_crameri_batlowK, 42
- scale_crameri_batlowW, 45
- scale_crameri_berlin, 48
- scale_crameri_bilbao, 51
- scale_crameri_broc, 54
- scale_crameri_broc0, 57
- scale_crameri_buda, 60
- scale_crameri_bukavu, 63
- scale_crameri_cork, 65
- scale_crameri_cork0, 68
- scale_crameri_davos, 71
- scale_crameri_devon, 74
- scale_crameri_fes, 77
- scale_crameri_grayC, 80
- scale_crameri_hawaii, 83
- scale_crameri_imola, 86
- scale_crameri_lajolla, 89
- scale_crameri_lapaz, 92
- scale_crameri_lisbon, 95
- scale_crameri_nuuk, 98
- scale_crameri_oleron, 102
- scale_crameri_oslo, 104
- scale_crameri_roma, 107
- scale_crameri_roma0, 110
- scale_crameri_tofino, 113
- scale_crameri_tokyo, 116
- scale_crameri_turku, 119
- scale_crameri_vanimo, 122
- scale_crameri_vik, 125
- scale_crameri_vik0, 128

* Okabe and Ito's color scheme

- scale_okabeito_discrete, 130

* Paul Tol's color schemes

- scale_tol_bright, 133
- scale_tol_BuRd, 135
- scale_tol_dark, 138
- scale_tol_discreterainbow, 140
- scale_tol_highcontrast, 142
- scale_tol_incandescent, 145
- scale_tol_iridescent, 147
- scale_tol_light, 150
- scale_tol_mediumcontrast, 152
- scale_tol_muted, 154
- scale_tol_nightfall, 156
- scale_tol_pale, 159
- scale_tol_PRGn, 161
- scale_tol_smoothrainbow, 163
- scale_tol_sunset, 166
- scale_tol_vibrant, 169
- scale_tol_YlOrBr, 171

* color schemes

- colour, 5
- info, 9

* color-blind safe color schemes

- scale_okabeito_discrete, 130

* color

- colour, 5

* cyclic color schemes

- scale_crameri_bam0, 36
- scale_crameri_broc0, 57
- scale_crameri_cork0, 68
- scale_crameri_roma0, 110
- scale_crameri_vik0, 128

* diagnostic tools

- change, 4
- compare, 8
- plot.color_scheme, 17
- plot_map, 18
- plot_scheme, 19
- plot_scheme_colourblind, 20
- plot_tiles, 21

* diverging color schemes

- scale_crameri_bam, 30
- scale_crameri_berlin, 48
- scale_crameri_broc, 54
- scale_crameri_cork, 65
- scale_crameri_lisbon, 95
- scale_crameri_roma, 107
- scale_crameri_tofino, 113
- scale_crameri_vanimo, 122
- scale_crameri_vik, 125
- scale_tol_BuRd, 135
- scale_tol_nightfall, 156
- scale_tol_PRGn, 161
- scale_tol_sunset, 166
- * **ggplot2 scales**
 - scale_picker, 132
- * **multi sequential color schemes**
 - scale_crameri_bukavu, 63
 - scale_crameri_fes, 77
 - scale_crameri_oleron, 102
- * **palettes**
 - palette_color_continuous, 10
 - palette_color_discrete, 11
 - palette_color_picker, 13
 - palette_shape, 14
 - palette_size, 15
- * **qualitative color schemes**
 - scale_colour_land, 22
 - scale_colour_soil, 24
 - scale_colour_stratigraphy, 25
 - scale_okabeito_discrete, 130
 - scale_tol_bright, 133
 - scale_tol_dark, 138
 - scale_tol_discreterainbow, 140
 - scale_tol_highcontrast, 142
 - scale_tol_light, 150
 - scale_tol_mediumcontrast, 152
 - scale_tol_muted, 154
 - scale_tol_pale, 159
 - scale_tol_vibrant, 169
- * **sequential color schemes**
 - scale_crameri_acton, 27
 - scale_crameri_bamako, 33
 - scale_crameri_batlow, 39
 - scale_crameri_batlowK, 42
 - scale_crameri_batlowW, 45
 - scale_crameri_bilbao, 51
 - scale_crameri_buda, 60
 - scale_crameri_davos, 71
 - scale_crameri_devon, 74
 - scale_crameri_grayC, 80
 - scale_crameri_hawaii, 83
 - scale_crameri_imola, 86
 - scale_crameri_lajolla, 89
 - scale_crameri_lapaz, 92
 - scale_crameri_nuuk, 98
 - scale_crameri_oslo, 104
 - scale_crameri_tokyo, 116
 - scale_crameri_turku, 119
 - scale_tol_incandescent, 145
 - scale_tol_iridescent, 147
 - scale_tol_smoothrainbow, 163
 - scale_tol_YlOrBr, 171
- * **themed color schemes**
 - scale_colour_land, 22
 - scale_colour_soil, 24
 - scale_colour_stratigraphy, 25
- change, 4, 8, 17–19, 21, 22
- character, 4, 6, 8, 10, 12–15, 17–21, 23, 24, 26, 28, 32, 35, 37, 40, 43, 46, 50, 53, 56, 58, 61, 64, 67, 69, 72, 75, 78, 81, 84, 87, 90, 94, 97, 100, 102, 105, 109, 111, 114, 117, 120, 123, 126, 129, 131, 133, 134, 137, 139, 141, 143, 146, 149, 150, 152, 154, 157, 159, 162, 165, 168, 169, 172
- color (colour), 5
- color(), 4, 13, 17
- colorRampPalette, 6
- colour, 5, 9
- compare, 4, 8, 17–19, 21, 22
- continuous, 29, 32, 35, 37, 40, 43, 46, 50, 53, 56, 58, 61, 64, 67, 69, 72, 75, 78, 81, 84, 87, 90, 94, 97, 100, 102, 106, 109, 111, 114, 117, 120, 123, 126, 129, 133, 137, 146, 149, 158, 162, 165, 168, 172
- data.frame, 9
- discrete, 23, 24, 26, 132–134, 139, 141, 143, 150, 152, 154, 159, 169
- distance matrix, 8
- function, 4, 6, 10, 12, 14–16
- ggplot2::continuous_scale(), 28, 32, 34, 37, 40, 43, 46, 50, 52, 56, 58, 61, 64,

- 67, 69, 72, 75, 78, 81, 84, 87, 90, 93,
97, 100, 102, 105, 109, 111, 114,
117, 120, 123, 126, 129, 137, 146,
148, 157, 162, 164, 167, 172
- ggplot2::discrete_scale(), 23, 24, 26,
131, 134, 139, 141, 143, 150, 152,
154, 159, 169
- graphics::par(), 19
- grDevices::colorRamp(), 10
- info, 7, 9
- info(), 6, 133
- integer, 6, 8, 21
- logical, 6, 8, 12–14, 19, 28, 32, 34, 35, 37,
40, 43, 46, 50, 52, 53, 56, 58, 61, 64,
67, 69, 72, 75, 78, 81, 84, 87, 90, 93,
94, 97, 100, 102, 105, 109, 111, 114,
117, 120, 123, 126, 129, 131, 134,
137, 139, 141, 143, 146, 148, 150,
152, 154, 157, 159, 162, 164,
167–169, 172
- numeric, 10, 13, 16, 19, 28, 32, 34, 37, 40, 43,
46, 50, 52, 56, 58, 61, 64, 67, 69, 72,
75, 78, 81, 84, 87, 90, 93, 97, 100,
102, 105, 109, 111, 114, 117, 120,
123, 126, 129, 137, 146, 148, 157,
162, 164, 168, 172
- palette_color_continuous, 10, 12, 14–16
- palette_color_discrete, 10, 11, 14–16
- palette_color_picker, 10, 12, 13, 15, 16
- palette_colour_continuous
(palette_color_continuous), 10
- palette_colour_discrete
(palette_color_discrete), 11
- palette_colour_picker
(palette_color_picker), 13
- palette_line (palette_shape), 14
- palette_shape, 10, 12, 14, 14, 16
- palette_size, 10, 12, 14, 15, 15
- palette_size_diverging (palette_size),
15
- palette_size_sequential (palette_size),
15
- plot.color_scheme, 4, 8, 17, 18, 19, 21, 22
- plot_map, 4, 8, 17, 18, 19, 21, 22
- plot_scheme, 4, 8, 17, 18, 19, 21, 22
- plot_scheme_colorblind
(plot_scheme_colourblind), 20
- plot_scheme_colourblind, 4, 8, 17–19, 20,
22
- plot_tiles, 4, 8, 17–19, 21, 21
- scale_color_acton
(scale_crameri_acton), 27
- scale_color_bam (scale_crameri_bam), 30
- scale_color_bamako
(scale_crameri_bamako), 33
- scale_color_bam0 (scale_crameri_bam0),
36
- scale_color_batlow
(scale_crameri_batlow), 39
- scale_color_batlowK
(scale_crameri_batlowK), 42
- scale_color_batlowW
(scale_crameri_batlowW), 45
- scale_color_berlin
(scale_crameri_berlin), 48
- scale_color_bilbao
(scale_crameri_bilbao), 51
- scale_color_bright (scale_tol_bright),
133
- scale_color_broc (scale_crameri_broc),
54
- scale_color_broc0
(scale_crameri_broc0), 57
- scale_color_buda (scale_crameri_buda),
60
- scale_color_bukavu
(scale_crameri_bukavu), 63
- scale_color_BuRd (scale_tol_BuRd), 135
- scale_color_cork (scale_crameri_cork),
65
- scale_color_cork0
(scale_crameri_cork0), 68
- scale_color_dark (scale_tol_dark), 138
- scale_color_davos
(scale_crameri_davos), 71
- scale_color_devon
(scale_crameri_devon), 74
- scale_color_discreterainbow
(scale_tol_discreterainbow),
140
- scale_color_fes (scale_crameri_fes), 77
- scale_color_grayC
(scale_crameri_grayC), 80

- scale_color_hawaii
(scale_crameri_hawaii), 83
- scale_color_highcontrast
(scale_tol_highcontrast), 142
- scale_color_imola
(scale_crameri_imola), 86
- scale_color_incandescent
(scale_tol_incandescent), 145
- scale_color_iridescent
(scale_tol_iridescent), 147
- scale_color_lajolla
(scale_crameri_lajolla), 89
- scale_color_land (scale_colour_land), 22
- scale_color_lapaz
(scale_crameri_lapaz), 92
- scale_color_light (scale_tol_light), 150
- scale_color_lisbon
(scale_crameri_lisbon), 95
- scale_color_mediumcontrast
(scale_tol_mediumcontrast), 152
- scale_color_muted (scale_tol_muted), 154
- scale_color_nightfall
(scale_tol_nightfall), 156
- scale_color_nuuk (scale_crameri_nuuk),
98
- scale_color_okabeito
(scale_okabeito_discrete), 130
- scale_color_oleron
(scale_crameri_oleron), 102
- scale_color_oslo (scale_crameri_oslo),
104
- scale_color_pale (scale_tol_pale), 159
- scale_color_picker (scale_picker), 132
- scale_color_PRGn (scale_tol_PRGn), 161
- scale_color_roma (scale_crameri_roma),
107
- scale_color_roma0
(scale_crameri_roma0), 110
- scale_color_smoothrainbow
(scale_tol_smoothrainbow), 163
- scale_color_soil (scale_colour_soil), 24
- scale_color_stratigraphy
(scale_colour_stratigraphy), 25
- scale_color_sunset (scale_tol_sunset),
166
- scale_color_tofino
(scale_crameri_tofino), 113
- scale_color_tokyo
(scale_crameri_tokyo), 116
- scale_color_turku
(scale_crameri_turku), 119
- scale_color_vanimo
(scale_crameri_vanimo), 122
- scale_color_vibrant
(scale_tol_vibrant), 169
- scale_color_vik (scale_crameri_vik), 125
- scale_color_vik0 (scale_crameri_vik0),
128
- scale_color_YlOrBr (scale_tol_YlOrBr),
171
- scale_colour_acton
(scale_crameri_acton), 27
- scale_colour_bam (scale_crameri_bam), 30
- scale_colour_bamako
(scale_crameri_bamako), 33
- scale_colour_bam0 (scale_crameri_bam0),
36
- scale_colour_batlow
(scale_crameri_batlow), 39
- scale_colour_batlowK
(scale_crameri_batlowK), 42
- scale_colour_batlowW
(scale_crameri_batlowW), 45
- scale_colour_berlin
(scale_crameri_berlin), 48
- scale_colour_bilbao
(scale_crameri_bilbao), 51
- scale_colour_bright (scale_tol_bright),
133
- scale_colour_broc (scale_crameri_broc),
54
- scale_colour_broc0
(scale_crameri_broc0), 57
- scale_colour_buda (scale_crameri_buda),
60
- scale_colour_bukavu
(scale_crameri_bukavu), 63
- scale_colour_BuRd (scale_tol_BuRd), 135
- scale_colour_cork (scale_crameri_cork),
65
- scale_colour_cork0
(scale_crameri_cork0), 68
- scale_colour_dark (scale_tol_dark), 138
- scale_colour_davos
(scale_crameri_davos), 71
- scale_colour_devon

- (scale_crameri_devon), 74
- scale_colour_discreterainbow
 - (scale_tol_discreterainbow), 140
- scale_colour_fes (scale_crameri_fes), 77
- scale_colour_grayC
 - (scale_crameri_grayC), 80
- scale_colour_hawaii
 - (scale_crameri_hawaii), 83
- scale_colour_highcontrast
 - (scale_tol_highcontrast), 142
- scale_colour_imola
 - (scale_crameri_imola), 86
- scale_colour_incandescent
 - (scale_tol_incandescent), 145
- scale_colour_iridescent
 - (scale_tol_iridescent), 147
- scale_colour_lajolla
 - (scale_crameri_lajolla), 89
- scale_colour_land, 22, 25, 26, 132, 135, 140, 142, 144, 151, 153, 155, 160, 170
- scale_colour_lapaz
 - (scale_crameri_lapaz), 92
- scale_colour_light (scale_tol_light), 150
- scale_colour_lisbon
 - (scale_crameri_lisbon), 95
- scale_colour_mediumcontrast
 - (scale_tol_mediumcontrast), 152
- scale_colour_muted (scale_tol_muted), 154
- scale_colour_nightfall
 - (scale_tol_nightfall), 156
- scale_colour_nuuk (scale_crameri_nuuk), 98
- scale_colour_okabeito
 - (scale_okabeito_discrete), 130
- scale_colour_oleron
 - (scale_crameri_oleron), 102
- scale_colour_oslo (scale_crameri_oslo), 104
- scale_colour_pale (scale_tol_pale), 159
- scale_colour_picker (scale_picker), 132
- scale_colour_PRGn (scale_tol_PRGn), 161
- scale_colour_roma (scale_crameri_roma), 107
- scale_colour_roma0
 - (scale_crameri_roma0), 110
- scale_colour_smoothrainbow
 - (scale_tol_smoothrainbow), 163
- scale_colour_soil, 23, 24, 26, 132, 135, 140, 142, 144, 151, 153, 155, 160, 170
- scale_colour_stratigraphy, 23, 25, 25, 132, 135, 140, 142, 144, 151, 153, 155, 160, 170
- scale_colour_sunset (scale_tol_sunset), 166
- scale_colour_tofino
 - (scale_crameri_tofino), 113
- scale_colour_tokyo
 - (scale_crameri_tokyo), 116
- scale_colour_turku
 - (scale_crameri_turku), 119
- scale_colour_vanimmo
 - (scale_crameri_vanimmo), 122
- scale_colour_vibrant
 - (scale_tol_vibrant), 169
- scale_colour_vik (scale_crameri_vik), 125
- scale_colour_vik0 (scale_crameri_vik0), 128
- scale_colour_YlOrBr (scale_tol_YlOrBr), 171
- scale_crameri_acton, 27, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_bam, 30, 30, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 138, 158, 163, 168
- scale_crameri_bamako, 30, 33, 33, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_bam0, 30, 33, 36, 36, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121,

- 124, 127, 130
- scale_crameri_batlow, 30, 33, 36, 38, 39, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_batlowK, 30, 33, 36, 38, 41, 42, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_batlowW, 30, 33, 36, 38, 41, 44, 45, 45, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_berlin, 30, 33, 36, 38, 41, 45, 48, 48, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 138, 158, 163, 168
- scale_crameri_bilbao, 30, 33, 36, 38, 41, 44, 45, 48, 51, 51, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_broc, 30, 33, 36, 38, 41, 45, 48, 51, 54, 54, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 138, 158, 163, 168
- scale_crameri_broc0, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 57, 62, 65, 68, 70, 73, 77, 79, 82, 85, 88, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130
- scale_crameri_buda, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 60, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_bukavu, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 63, 68, 70, 73, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130
- scale_crameri_cork, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 65, 70, 73, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130
- scale_crameri_cork0, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 68, 73, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130
- scale_crameri_davos, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 71, 76, 77, 79, 82, 85, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_devon, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 74, 79, 82, 85, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_fes, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 77, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130
- scale_crameri_grayC, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 80, 85, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_hawaii, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 83, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 127, 130, 147, 149, 166, 173
- scale_crameri_imola, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 86, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 125, 128, 130, 147, 149, 166,

- 173
- scale_crameri_lajolla, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 89, 89, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 125, 128, 130, 147, 149, 166, 173
- scale_crameri_lapaz, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 89, 92, 92, 98, 101, 104, 107, 110, 112, 115, 118, 121, 125, 128, 130, 147, 149, 166, 173
- scale_crameri_lisbon, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 89, 92, 95, 95, 101, 104, 107, 110, 112, 115, 118, 121, 124, 125, 127, 128, 130, 138, 158, 163, 168
- scale_crameri_nuuk, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 89, 92, 95, 98, 98, 104, 107, 110, 112, 115, 118, 121, 125, 128, 130, 147, 149, 166, 173
- scale_crameri_oleron, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 89, 92, 95, 98, 101, 102, 107, 110, 112, 115, 118, 121, 125, 128, 130
- scale_crameri_oslo, 30, 33, 36, 38, 41, 44, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 76, 77, 79, 82, 85, 88, 89, 92, 95, 98, 101, 104, 104, 110, 112, 115, 118, 121, 125, 128, 130, 147, 149, 166, 173
- scale_crameri_roma, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 107, 112, 115, 118, 121, 124, 125, 127, 128, 130, 138, 158, 163, 168
- scale_crameri_roma0, 30, 33, 36, 38, 41, 45, 48, 51, 54, 57, 59, 62, 65, 68, 70, 73, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 110, 115, 118, 121, 125, 128, 130
- scale_crameri_tofino, 30, 33, 36, 38, 42, 45, 48, 51, 54, 57, 59, 63, 65, 68, 70, 74, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 113, 118, 121, 124, 125, 127, 128, 130, 138, 158, 163, 168
- scale_crameri_tokyo, 30, 33, 36, 38, 41, 42, 44, 45, 48, 51, 54, 57, 59, 62, 63, 65, 68, 70, 73, 74, 76, 77, 79, 82, 85, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 112, 115, 116, 121, 125, 128, 130, 147, 149, 166, 173
- scale_crameri_turku, 30, 33, 36, 38, 41, 42, 44, 45, 48, 51, 54, 57, 59, 62, 63, 65, 68, 70, 73, 74, 76, 77, 79, 82, 85, 88, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 119, 125, 128, 130, 147, 149, 166, 173
- scale_crameri_vanimo, 30, 33, 36, 38, 42, 45, 48, 51, 54, 57, 59, 63, 65, 68, 70, 74, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 122, 127, 128, 130, 138, 158, 163, 168
- scale_crameri_vik, 30, 33, 36, 38, 42, 45, 48, 51, 54, 57, 59, 63, 65, 68, 70, 74, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 124, 125, 125, 130, 138, 158, 163, 168
- scale_crameri_vik0, 30, 33, 36, 38, 42, 45, 48, 51, 54, 57, 59, 63, 65, 68, 70, 74, 77, 79, 82, 85, 89, 92, 95, 98, 101, 104, 107, 110, 112, 115, 118, 121, 125, 128, 128
- scale_edge_color_acton
(scale_crameri_acton), 27
- scale_edge_color_bam
(scale_crameri_bam), 30
- scale_edge_color_bamako
(scale_crameri_bamako), 33
- scale_edge_color_batlow
(scale_crameri_batlow), 39
- scale_edge_color_batlowK
(scale_crameri_batlowK), 42
- scale_edge_color_batlowW
(scale_crameri_batlowW), 45
- scale_edge_color_berlin
(scale_crameri_berlin), 48

- scale_edge_color_bilbao
 - (scale_crameri_bilbao), 51
- scale_edge_color_bright
 - (scale_tol_bright), 133
- scale_edge_color_broc
 - (scale_crameri_broc), 54
- scale_edge_color_buda
 - (scale_crameri_buda), 60
- scale_edge_color_BuRd (scale_tol_BuRd), 135
- scale_edge_color_cork
 - (scale_crameri_cork), 65
- scale_edge_color_dark (scale_tol_dark), 138
- scale_edge_color_davos
 - (scale_crameri_davos), 71
- scale_edge_color_devon
 - (scale_crameri_devon), 74
- scale_edge_color_discreterainbow
 - (scale_tol_discreterainbow), 140
- scale_edge_color_grayC
 - (scale_crameri_grayC), 80
- scale_edge_color_hawaii
 - (scale_crameri_hawaii), 83
- scale_edge_color_highcontrast
 - (scale_tol_highcontrast), 142
- scale_edge_color_imola
 - (scale_crameri_imola), 86
- scale_edge_color_incandescent
 - (scale_tol_incandescent), 145
- scale_edge_color_iridescent
 - (scale_tol_iridescent), 147
- scale_edge_color_lajolla
 - (scale_crameri_lajolla), 89
- scale_edge_color_land
 - (scale_colour_land), 22
- scale_edge_color_lapaz
 - (scale_crameri_lapaz), 92
- scale_edge_color_light
 - (scale_tol_light), 150
- scale_edge_color_lisbon
 - (scale_crameri_lisbon), 95
- scale_edge_color_mediumcontrast
 - (scale_tol_mediumcontrast), 152
- scale_edge_color_muted
 - (scale_tol_muted), 154
- scale_edge_color_nightfall
 - (scale_tol_nightfall), 156
- scale_edge_color_nuuk
 - (scale_crameri_nuuk), 98
- scale_edge_color_okabeito
 - (scale_okabeito_discrete), 130
- scale_edge_color_oslo
 - (scale_crameri_oslo), 104
- scale_edge_color_pale (scale_tol_pale), 159
- scale_edge_color_picker (scale_picker), 132
- scale_edge_color_PRGn (scale_tol_PRGn), 161
- scale_edge_color_roma
 - (scale_crameri_roma), 107
- scale_edge_color_smoothrainbow
 - (scale_tol_smoothrainbow), 163
- scale_edge_color_soil
 - (scale_colour_soil), 24
- scale_edge_color_stratigraphy
 - (scale_colour_stratigraphy), 25
- scale_edge_color_sunset
 - (scale_tol_sunset), 166
- scale_edge_color_tofino
 - (scale_crameri_tofino), 113
- scale_edge_color_tokyo
 - (scale_crameri_tokyo), 116
- scale_edge_color_turku
 - (scale_crameri_turku), 119
- scale_edge_color_vanimo
 - (scale_crameri_vanimo), 122
- scale_edge_color_vibrant
 - (scale_tol_vibrant), 169
- scale_edge_color_vik
 - (scale_crameri_vik), 125
- scale_edge_color_YlOrBr
 - (scale_tol_YlOrBr), 171
- scale_edge_colour_acton
 - (scale_crameri_acton), 27
- scale_edge_colour_bam
 - (scale_crameri_bam), 30
- scale_edge_colour_bamako
 - (scale_crameri_bamako), 33
- scale_edge_colour_batlow
 - (scale_crameri_batlow), 39
- scale_edge_colour_batlowK
 - (scale_crameri_batlowK), 42
- scale_edge_colour_batlowW

- (scale_crameri_batlowW), 45
- scale_edge_colour_berlin
 - (scale_crameri_berlin), 48
- scale_edge_colour_bilbao
 - (scale_crameri_bilbao), 51
- scale_edge_colour_bright
 - (scale_tol_bright), 133
- scale_edge_colour_broc
 - (scale_crameri_broc), 54
- scale_edge_colour_buda
 - (scale_crameri_buda), 60
- scale_edge_colour_BuRd
 - (scale_tol_BuRd), 135
- scale_edge_colour_cork
 - (scale_crameri_cork), 65
- scale_edge_colour_dark
 - (scale_tol_dark), 138
- scale_edge_colour_davos
 - (scale_crameri_davos), 71
- scale_edge_colour_devon
 - (scale_crameri_devon), 74
- scale_edge_colour_discreterainbow
 - (scale_tol_discreterainbow), 140
- scale_edge_colour_grayC
 - (scale_crameri_grayC), 80
- scale_edge_colour_hawaii
 - (scale_crameri_hawaii), 83
- scale_edge_colour_highcontrast
 - (scale_tol_highcontrast), 142
- scale_edge_colour_imola
 - (scale_crameri_imola), 86
- scale_edge_colour_incandescent
 - (scale_tol_incandescent), 145
- scale_edge_colour_iridescent
 - (scale_tol_iridescent), 147
- scale_edge_colour_lajolla
 - (scale_crameri_lajolla), 89
- scale_edge_colour_land
 - (scale_colour_land), 22
- scale_edge_colour_lapaz
 - (scale_crameri_lapaz), 92
- scale_edge_colour_light
 - (scale_tol_light), 150
- scale_edge_colour_lisbon
 - (scale_crameri_lisbon), 95
- scale_edge_colour_mediumcontrast
 - (scale_tol_mediumcontrast), 152
- scale_edge_colour_muted
 - (scale_tol_muted), 154
- scale_edge_colour_nightfall
 - (scale_tol_nightfall), 156
- scale_edge_colour_nuuk
 - (scale_crameri_nuuk), 98
- scale_edge_colour_okabeito
 - (scale_okabeito_discrete), 130
- scale_edge_colour_oslo
 - (scale_crameri_oslo), 104
- scale_edge_colour_pale
 - (scale_tol_pale), 159
- scale_edge_colour_picker
 - (scale_picker), 132
- scale_edge_colour_PRGn
 - (scale_tol_PRGn), 161
- scale_edge_colour_roma
 - (scale_crameri_roma), 107
- scale_edge_colour_smoothrainbow
 - (scale_tol_smoothrainbow), 163
- scale_edge_colour_soil
 - (scale_colour_soil), 24
- scale_edge_colour_stratigraphy
 - (scale_colour_stratigraphy), 25
- scale_edge_colour_sunset
 - (scale_tol_sunset), 166
- scale_edge_colour_tofino
 - (scale_crameri_tofino), 113
- scale_edge_colour_tokyo
 - (scale_crameri_tokyo), 116
- scale_edge_colour_turku
 - (scale_crameri_turku), 119
- scale_edge_colour_vanimo
 - (scale_crameri_vanimo), 122
- scale_edge_colour_vibrant
 - (scale_tol_vibrant), 169
- scale_edge_colour_vik
 - (scale_crameri_vik), 125
- scale_edge_colour_YlOrBr
 - (scale_tol_YlOrBr), 171
- scale_edge_fill_acton
 - (scale_crameri_acton), 27
- scale_edge_fill_bam
 - (scale_crameri_bam), 30
- scale_edge_fill_bamako
 - (scale_crameri_bamako), 33
- scale_edge_fill_batlow
 - (scale_crameri_batlow), 39

- scale_edge_fill_batlowK
(scale_crameri_batlowK), 42
- scale_edge_fill_batlowW
(scale_crameri_batlowW), 45
- scale_edge_fill_berlin
(scale_crameri_berlin), 48
- scale_edge_fill_bilbao
(scale_crameri_bilbao), 51
- scale_edge_fill_bright
(scale_tol_bright), 133
- scale_edge_fill_broc
(scale_crameri_broc), 54
- scale_edge_fill_buda
(scale_crameri_buda), 60
- scale_edge_fill_BuRd (scale_tol_BuRd),
135
- scale_edge_fill_cork
(scale_crameri_cork), 65
- scale_edge_fill_dark (scale_tol_dark),
138
- scale_edge_fill_davos
(scale_crameri_davos), 71
- scale_edge_fill_devon
(scale_crameri_devon), 74
- scale_edge_fill_discreterainbow
(scale_tol_discreterainbow),
140
- scale_edge_fill_grayC
(scale_crameri_grayC), 80
- scale_edge_fill_hawaii
(scale_crameri_hawaii), 83
- scale_edge_fill_highcontrast
(scale_tol_highcontrast), 142
- scale_edge_fill_imola
(scale_crameri_imola), 86
- scale_edge_fill_incandescent
(scale_tol_incandescent), 145
- scale_edge_fill_iridescent
(scale_tol_iridescent), 147
- scale_edge_fill_lajolla
(scale_crameri_lajolla), 89
- scale_edge_fill_land
(scale_colour_land), 22
- scale_edge_fill_lapaz
(scale_crameri_lapaz), 92
- scale_edge_fill_light
(scale_tol_light), 150
- scale_edge_fill_lisbon
(scale_crameri_lisbon), 95
- scale_edge_fill_mediumcontrast
(scale_tol_mediumcontrast), 152
- scale_edge_fill_muted
(scale_tol_muted), 154
- scale_edge_fill_nightfall
(scale_tol_nightfall), 156
- scale_edge_fill_nuuk
(scale_crameri_nuuk), 98
- scale_edge_fill_okabeito
(scale_okabeito_discrete), 130
- scale_edge_fill_oslo
(scale_crameri_oslo), 104
- scale_edge_fill_pale (scale_tol_pale),
159
- scale_edge_fill_picker (scale_picker),
132
- scale_edge_fill_PRGn (scale_tol_PRGn),
161
- scale_edge_fill_roma
(scale_crameri_roma), 107
- scale_edge_fill_smoothrainbow
(scale_tol_smoothrainbow), 163
- scale_edge_fill_soil
(scale_colour_soil), 24
- scale_edge_fill_stratigraphy
(scale_colour_stratigraphy), 25
- scale_edge_fill_sunset
(scale_tol_sunset), 166
- scale_edge_fill_tofino
(scale_crameri_tofino), 113
- scale_edge_fill_tokyo
(scale_crameri_tokyo), 116
- scale_edge_fill_turku
(scale_crameri_turku), 119
- scale_edge_fill_vanimo
(scale_crameri_vanimo), 122
- scale_edge_fill_vibrant
(scale_tol_vibrant), 169
- scale_edge_fill_vik
(scale_crameri_vik), 125
- scale_edge_fill_YlOrBr
(scale_tol_YlOrBr), 171
- scale_fill_acton (scale_crameri_acton),
27
- scale_fill_bam (scale_crameri_bam), 30
- scale_fill_bamako
(scale_crameri_bamako), 33

- scale_fill_bam0 (scale_crameri_bam0), 36
- scale_fill_batlow
 - (scale_crameri_batlow), 39
- scale_fill_batlowK
 - (scale_crameri_batlowK), 42
- scale_fill_batlowW
 - (scale_crameri_batlowW), 45
- scale_fill_berlin
 - (scale_crameri_berlin), 48
- scale_fill_bilbao
 - (scale_crameri_bilbao), 51
- scale_fill_bright (scale_tol_bright), 133
- scale_fill_broc (scale_crameri_broc), 54
- scale_fill_broc0 (scale_crameri_broc0), 57
- scale_fill_buda (scale_crameri_buda), 60
- scale_fill_bukavu
 - (scale_crameri_bukavu), 63
- scale_fill_BuRd (scale_tol_BuRd), 135
- scale_fill_cork (scale_crameri_cork), 65
- scale_fill_cork0 (scale_crameri_cork0), 68
- scale_fill_dark (scale_tol_dark), 138
- scale_fill_davos (scale_crameri_davos), 71
- scale_fill_devon (scale_crameri_devon), 74
- scale_fill_discreterainbow
 - (scale_tol_discreterainbow), 140
- scale_fill_fes (scale_crameri_fes), 77
- scale_fill_grayC (scale_crameri_grayC), 80
- scale_fill_hawaii
 - (scale_crameri_hawaii), 83
- scale_fill_highcontrast
 - (scale_tol_highcontrast), 142
- scale_fill_imola (scale_crameri_imola), 86
- scale_fill_incandescent
 - (scale_tol_incandescent), 145
- scale_fill_iridescent
 - (scale_tol_iridescent), 147
- scale_fill_lajolla
 - (scale_crameri_lajolla), 89
- scale_fill_land (scale_colour_land), 22
- scale_fill_lapaz (scale_crameri_lapaz), 92
- scale_fill_light (scale_tol_light), 150
- scale_fill_lisbon
 - (scale_crameri_lisbon), 95
- scale_fill_mediumcontrast
 - (scale_tol_mediumcontrast), 152
- scale_fill_muted (scale_tol_muted), 154
- scale_fill_nightfall
 - (scale_tol_nightfall), 156
- scale_fill_nuuk (scale_crameri_nuuk), 98
- scale_fill_okabeito
 - (scale_okabeito_discrete), 130
- scale_fill_oleron
 - (scale_crameri_oleron), 102
- scale_fill_oslo (scale_crameri_oslo), 104
- scale_fill_pale (scale_tol_pale), 159
- scale_fill_picker (scale_picker), 132
- scale_fill_PRGn (scale_tol_PRGn), 161
- scale_fill_roma (scale_crameri_roma), 107
- scale_fill_roma0 (scale_crameri_roma0), 110
- scale_fill_smoothrainbow
 - (scale_tol_smoothrainbow), 163
- scale_fill_soil (scale_colour_soil), 24
- scale_fill_stratigraphy
 - (scale_colour_stratigraphy), 25
- scale_fill_sunset (scale_tol_sunset), 166
- scale_fill_tofino
 - (scale_crameri_tofino), 113
- scale_fill_tokyo (scale_crameri_tokyo), 116
- scale_fill_turku (scale_crameri_turku), 119
- scale_fill_vanimo
 - (scale_crameri_vanimo), 122
- scale_fill_vibrant (scale_tol_vibrant), 169
- scale_fill_vik (scale_crameri_vik), 125
- scale_fill_vik0 (scale_crameri_vik0), 128
- scale_fill_YlOrBr (scale_tol_YlOrBr), 171
- scale_okabeito_discrete, 23, 25, 26, 130, 135, 140, 142, 144, 151, 153, 155, 160, 170

- scale_picker*, 132
scale_tol_bright, 23, 25, 26, 132, 133, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_BuRd, 33, 51, 57, 68, 98, 110, 115, 124, 127, 135, 135, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_dark, 23, 25, 26, 132, 135, 138, 138, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_discreterainbow, 23, 25, 26, 132, 135, 138, 140, 140, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_highcontrast, 23, 25, 26, 132, 135, 138, 140, 142, 142, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_incandescent, 30, 36, 41, 44, 48, 54, 62, 73, 76, 82, 85, 88, 92, 95, 101, 107, 118, 121, 135, 138, 140, 142, 144, 145, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_iridescent, 30, 36, 41, 44, 48, 54, 62, 73, 76, 82, 85, 88, 92, 95, 101, 107, 118, 121, 135, 138, 140, 142, 144, 147, 147, 151, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_light, 23, 25, 26, 132, 135, 138, 140, 142, 144, 147, 149, 150, 153, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_mediumcontrast, 23, 25, 26, 132, 135, 138, 140, 142, 144, 147, 149, 151, 152, 155, 158, 160, 163, 166, 168, 170, 173
scale_tol_muted, 23, 25, 26, 132, 135, 138, 140, 142, 144, 147, 149, 151, 153, 154, 158, 160, 163, 166, 168, 170, 173
scale_tol_nightfall, 33, 51, 57, 68, 98, 110, 115, 124, 127, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 156, 160, 163, 166, 168, 170, 173
scale_tol_pale, 23, 25, 26, 132, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 159, 163, 166, 168, 170, 173
scale_tol_PRGn, 33, 51, 57, 68, 98, 110, 115, 124, 127, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 161, 166, 168, 170, 173
scale_tol_smoothrainbow, 30, 36, 41, 44, 48, 54, 62, 73, 76, 82, 85, 88, 92, 95, 101, 107, 118, 121, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 163, 168, 170, 173
scale_tol_sunset, 33, 51, 57, 68, 98, 110, 115, 124, 127, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 166, 170, 173
scale_tol_vibrant, 23, 25, 26, 132, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 169, 173
scale_tol_YlOrBr, 30, 36, 41, 44, 48, 54, 62, 73, 76, 82, 85, 88, 92, 95, 101, 107, 118, 121, 135, 138, 140, 142, 144, 147, 149, 151, 153, 155, 158, 160, 163, 166, 168, 170, 171
spacesXYZ::DeltaE(), 8