

# Package ‘khroma’

January 8, 2024

**Title** Colour Schemes for Scientific Data Visualization

**Version** 1.12.0

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

**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://packages.tesselle.org/khroma/>,  
<https://github.com/tesselle/khroma>

**BugReports** <https://github.com/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** 7.2.3

**Collate** 'anomalize.R' 'color.R' 'compare.R' 'convert.R' 'ggplot2.R'  
'info.R' 'khroma-internal.R' 'khroma-package.R' 'plot.R'  
'plot\_map.R' 'plot\_scheme.R' 'plot\_scheme\_colorblind.R'  
'plot\_tiles.R' 'print.R' 'ramp.R' 'scale\_colour\_crameri.R'  
'scale\_colour\_okabeito.R' 'scale\_colour\_other.R'  
'scale\_colour\_science.R' 'scale\_colour\_tol.R'

**NeedsCompilation** no

**Author** Nicolas Frerebeau [aut, cre] (<<https://orcid.org/0000-0001-5759-4944>>, Université Bordeaux Montaigne),  
 Brice Lebrun [ctb] (<<https://orcid.org/0000-0001-7503-8685>>, Université Bordeaux Montaigne),  
 Vincent Arel-Bundock [ctb] (<<https://orcid.org/0000-0003-2042-7063>>, Université de Montréal),  
 Ulrik Stervbo [ctb] (<<https://orcid.org/0000-0002-2831-8868>>, Ruhr-Universität Bochum),  
 Université Bordeaux Montaigne [fnd],  
 CNRS [fnd]

**Repository** CRAN

**Date/Publication** 2024-01-08 12:20:02 UTC

## R topics documented:

colour . . . . .	3
compare . . . . .	7
convert . . . . .	8
info . . . . .	9
plot.color_scheme . . . . .	10
plot_map . . . . .	11
plot_scheme . . . . .	12
plot_scheme_colourblind . . . . .	13
plot_tiles . . . . .	14
ramp . . . . .	15
scale_colour_land . . . . .	16
scale_colour_soil . . . . .	18
scale_colour_stratigraphy . . . . .	19
scale_crameri_acton . . . . .	21
scale_crameri_bam . . . . .	23
scale_crameri_bamako . . . . .	26
scale_crameri_bamO . . . . .	29
scale_crameri_batlow . . . . .	31
scale_crameri_batlowK . . . . .	33
scale_crameri_batlowW . . . . .	36
scale_crameri_berlin . . . . .	39
scale_crameri_bilbao . . . . .	41
scale_crameri_broc . . . . .	44
scale_crameri_brocO . . . . .	47
scale_crameri_buda . . . . .	49
scale_crameri_bukavu . . . . .	51
scale_crameri_cork . . . . .	53
scale_crameri_corkO . . . . .	56
scale_crameri_davos . . . . .	58
scale_crameri_devon . . . . .	60
scale_crameri_fes . . . . .	63
scale_crameri_grayC . . . . .	65

scale_crameri_hawaii . . . . .	68
scale_crameri_imola . . . . .	70
scale_crameri_lajolla . . . . .	73
scale_crameri_lapaz . . . . .	76
scale_crameri_lisbon . . . . .	78
scale_crameri_nuuk . . . . .	81
scale_crameri_oleron . . . . .	84
scale_crameri_oslo . . . . .	86
scale_crameri_roma . . . . .	88
scale_crameri_romaO . . . . .	91
scale_crameri_tofino . . . . .	93
scale_crameri_tokyo . . . . .	95
scale_crameri_turku . . . . .	98
scale_crameri_vanimo . . . . .	101
scale_crameri_vik . . . . .	103
scale_crameri_vikO . . . . .	106
scale_logical_discrete . . . . .	108
scale_okabeito_discrete . . . . .	109
scale_picker . . . . .	111
scale_tol_bright . . . . .	112
scale_tol_BuRd . . . . .	113
scale_tol_dark . . . . .	116
scale_tol_discreterainbow . . . . .	118
scale_tol_highcontrast . . . . .	120
scale_tol_incandescent . . . . .	122
scale_tol_iridescent . . . . .	124
scale_tol_light . . . . .	127
scale_tol_mediumcontrast . . . . .	128
scale_tol_muted . . . . .	130
scale_tol_nightfall . . . . .	132
scale_tol_pale . . . . .	135
scale_tol_PRGn . . . . .	137
scale_tol_smoothrainbow . . . . .	139
scale_tol_sunset . . . . .	142
scale_tol_vibrant . . . . .	144
scale_tol_YIOrBr . . . . .	146

**Index****149**

colour

*Color Palette***Description**

Provides qualitative, diverging and sequential color schemes.

**Usage**

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

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

**Arguments**

palette	A <a href="#">character</a> string giving the name of the palette to be used (see below).
reverse	A <a href="#">logical</a> scalar: should the resulting vector of colors should be reversed?
names	A <a href="#">logical</a> scalar: should the names of the colors should be kept in the resulting vector?
lang	A <a href="#">character</a> string specifying the language for the color names. It must be one of "en" (English, the default) or "fr" (French).
force	A <a href="#">logical</a> 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 <a href="#">colorRampPalette</a> .

**Value**

A palette function with the following attributes, that when called with a single integer argument (the number of levels) 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).

**Paul Tol's Color Schemes**

The following palettes are available. The maximum number of supported colors is in brackets, this value is only relevant for the qualitative color schemes (divergent and sequential schemes are linearly interpolated).

**Qualitative data** bright (7), high contrast (3), vibrant (7), muted (9), medium contrast (6), pale (6), dark (6), light (9).

**Diverging data** sunset (11), nightfall (17), BuRd (9), PRGn (9).

**Sequential data** YlOrBr (9), iridescent (23), incandescent (11), discrete rainbow (23), smooth rainbow (34).

### Qualitative Color Schemes

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).

### 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.

### Okabe and Ito Color Scheme

The following (qualitative) color scheme is available:

`okabeito` Up to 8 colors.

`okabeito black` Up to 8 colors, with black as the last.

### Scientific Color Schemes

The following (qualitative) color schemes are available:

`stratigraphy` International Chronostratigraphic Chart (175 colors).

`land` AVHRR Global Land Cover Classification (14 colors).

`soil` FAO Reference Soil Groups (24 colors).

### 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.

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 palettes: [info\(\)](#), [ramp\(\)](#), [scale\\_picker](#)

## 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 <a href="#">character</a> vector of colors.
metric	An <a href="#">integer</a> value giving the year the metric was recommended by the CIE. It must be one of "1976", "1994", or "2000" (default; see <a href="#">spacesXYZ::DeltaE()</a> ).
diag	A <a href="#">logical</a> scalar: should the diagonal of the distance matrix be printed?
upper	A <a href="#">logical</a> 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: [convert\(\)](#), [plot.color\\_scheme\(\)](#), [plot\\_map\(\)](#), [plot\\_scheme\\_colourblind\(\)](#), [plot\\_scheme\(\)](#), [plot\\_tiles\(\)](#)

## Examples

```
# Trichromat
pal <- colour("bright")

(deltaE <- compare(pal(5)))
summary(deltaE)

# Deuteranopia
deu <- convert(pal, mode = "deuteranopia")
compare(deu(5))
```

```
# Protanopia
pro <- convert(pal, mode = "protanopia")
compare(pro(5))

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

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

---

convert

*Simulate Color-Blindness*

---

## Description

Simulate Color-Blindness

## Usage

```
convert(x, mode)
```

## Arguments

x	A palette <a href="#">function</a> that when called with a single integer argument (the number of levels) returns a vector of colors (see <a href="#">color()</a> ).
mode	A <a href="#">character</a> 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\\_colourblind\(\)](#), [plot\\_scheme\(\)](#), [plot\\_tiles\(\)](#)

### Examples

```
# Trichromat
pal <- colour("bright")
plot_scheme(pal(7))

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

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

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

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

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

---

info

*Available Palettes*

---

### Description

Returns information about the available palettes.

### Usage

```
info()
```

**Value**

A `data.frame` with the following columns:

`palette` Names of palette.

`type` Types of palettes: sequential, diverging or qualitative.

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

`missing` The hexadecimal color value for mapping missing values.

**Author(s)**

N. Frerebeau

**See Also**

Other color palettes: `colour()`, `ramp()`, `scale_picker`

**Examples**

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

---

`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.

**Author(s)**

N. Frerebeau

**See Also**

Other diagnostic tools: `compare()`, `convert()`, `plot_map()`, `plot_scheme_colourblind()`, `plot_scheme()`, `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.

## Author(s)

N. Frerebeau, V. Arel-Bundock

## See Also

Other diagnostic tools: [compare\(\)](#), [convert\(\)](#), [plot.color\\_scheme\(\)](#), [plot\\_scheme\\_colourblind\(\)](#), [plot\\_scheme\(\)](#), [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 <a href="#">character</a> vector of colors.
colours	A <a href="#">logical</a> scalar: should the hexadecimal representation of the colors be displayed?
names	A <a href="#">logical</a> scalar: should the name of the colors be displayed?
size	A <a href="#">numeric</a> value giving the amount by which plotting text should be magnified relative to the default. Works the same as cex parameter of <a href="#">graphics::par()</a> .

**Author(s)**

N. Frerebeau

**See Also**

Other diagnostic tools: [compare\(\)](#), [convert\(\)](#), [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.

## Author(s)

N. Frerebeau, V. Arel-Bundock

## See Also

Other diagnostic tools: [compare\(\)](#), [convert\(\)](#), [plot.color\\_scheme\(\)](#), [plot\\_map\(\)](#), [plot\\_scheme\(\)](#), [plot\\_tiles\(\)](#)

## Examples

```
# Trichromat
pal <- colour("bright")
plot_scheme(pal(7))

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

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

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

# Achromatopsia
ach <- convert(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).

## Author(s)

N. Frerebeau

## See Also

Other diagnostic tools: [compare\(\)](#), [convert\(\)](#), [plot.color\\_scheme\(\)](#), [plot\\_map\(\)](#), [plot\\_scheme\\_colourblind\(\)](#), [plot\\_scheme\(\)](#)

## 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)
```

---

ramp

*Color Ramp*

---

## Description

Provides a color scheme that map a numeric vector to colors.

## Usage

```
ramp(palette, ...)
```

## Arguments

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

`...` Further arguments to be passed to [color\(\)](#).

## Value

A palette [function](#) with argument a vector of values that are mapped to colors. The data midpoint can be specified with an additional midpoint argument (see examples).

## Author(s)

N. Frerebeau

## See Also

Other color palettes: [colour\(\)](#), [info\(\)](#), [scale\\_picker](#)

**Examples**

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

## Where are breaks?
elevation$breaks

## Build palette function
ramp_BuRd <- ramp(palette = "BuRd")

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

## Rescale to midpoint
(col <- ramp_BuRd(elevation$breaks, midpoint = 160))
image(volcano, col = col)
legend("topright", legend = attr(col, "breaks"), fill = col)
```

---

scale_colour_land	<i>AVHRR Global Land Cover Classification Color Scheme for <b>ggplot2</b> and <b>ggraph</b></i>
-------------------	---

---

**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 <a href="#">character</a> 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\\_logical\\_discrete](#), [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 <a href="#">character</a> 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 <a href="#">character</a> 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\\_logical\\_discrete](#), [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 <a href="#">character</a> 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 <a href="#">character</a> 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\\_logical\\_discrete](#), [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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> scalar: should the color scheme be used as a discrete scale?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_bamako](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

## Value

A **continuous** scale.

## Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

## 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\\_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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

**Value**

A [continuous](#) scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

**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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.

discrete	A <a href="#">logical</a> scalar: should the color scheme be used as a discrete scale?
aesthetics	A <a href="#">character</a> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A [continuous](#) scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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_batlowW`, `scale_crameri_batlow`, `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_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `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_roma0,
scale_crameri_roma, scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku,
scale_crameri_vanimo, scale_crameri_vik0, scale_crameri_vik
```

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_batlowK](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

### 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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

## 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 <a href="#">logical</a> scalar. Should the resulting vector of colors be reversed?
range	A length-two <a href="#">numeric</a> vector specifying the fraction of the scheme's color do-

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

**Value**

A [continuous](#) scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

## Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

## 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\\_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\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [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\_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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> scalar: should the color scheme be used as a discrete scale?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

## Value

A **continuous** scale.

## Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

## 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\\_cork0](#), [scale\\_crameri\\_roma0](#), [scale\\_crameri\\_vik0](#)

Other Fabio Crameri's color schemes: [scale\\_crameri\\_acton](#), [scale\\_crameri\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [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\_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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> scalar: should the color scheme be used as a discrete scale?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `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_roma0`, `scale_crameri_roma`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimo`, `scale_crameri_vik0`, `scale_crameri_vik`

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

**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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [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\_roma0,  
 scale\_crameri\_roma, scale\_crameri\_tofino, scale\_crameri\_tokyo, scale\_crameri\_turku,  
 scale\_crameri\_vanimo, scale\_crameri\_vik0, scale\_crameri\_vik

## 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\_corkO    *Fabio Crameri's corkO Cyclic Color Scheme*

---

## Description

Fabio Crameri's *corkO* Cyclic Color Scheme

## Usage

```
scale_colour_cork0(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_color_cork0(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "colour"
)

scale_fill_cork0(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)
```



**Arguments**

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_roma0](#), [scale\\_crameri\\_vik0](#)

Other Fabio Crameri's color schemes: [scale\\_crameri\\_acton](#), [scale\\_crameri\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [scale\\_crameri\\_cork](#), [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\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `scale_crameri_cork`, `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_roma0,
scale_crameri_roma, scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku,
scale_crameri_vanimo, scale_crameri_vik0, scale_crameri_vik
```

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

## Value

A **continuous** scale.

## Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

## 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\\_oleron](#)

Other Fabio Crameri's color schemes: [scale\\_crameri\\_acton](#), [scale\\_crameri\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#),



```
scale_crameri_buda, scale_crameri_bukavu, scale_crameri_cork0, scale_crameri_cork,
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_roma0,
scale_crameri_roma, scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku,
scale_crameri_vanimo, scale_crameri_vik0, scale_crameri_vik
```

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [scale\\_crameri\\_cork](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `scale_crameri_cork`, `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_roma0`, `scale_crameri_roma`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimo`, `scale_crameri_vik0`, `scale_crameri_vik`

**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 <a href="#">logical</a> scalar. Should the resulting vector of colors be reversed?
range	A length-two <a href="#">numeric</a> vector specifying the fraction of the scheme's color do-

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

**Value**

A [continuous](#) scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [scale\\_crameri\\_cork](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimu](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)



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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [scale\\_crameri\\_cork](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `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_lisbon`, `scale_crameri_nuuk`, `scale_crameri_oleron`, `scale_crameri_oslo`, `scale_crameri_roma0`, `scale_crameri_roma`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimo`, `scale_crameri_vik0`, `scale_crameri_vik`

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

**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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_nuuk](#),



[scale\\_crameri\\_oleron](#), [scale\\_crameri\\_oslo](#), [scale\\_crameri\\_roma0](#), [scale\\_crameri\\_roma](#),  
[scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#),  
[scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_oleron](#), [scale\\_crameri\\_oslo](#), [scale\\_crameri\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimu](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

## Value

A **continuous** scale.

## Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

## 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\\_fes](#)

Other Fabio Crameri's color schemes: [scale\\_crameri\\_acton](#), [scale\\_crameri\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_oslo](#), [scale\\_crameri\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `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_roma0`, `scale_crameri_roma`, `scale_crameri_tofino`, `scale_crameri_tokyo`, `scale_crameri_turku`, `scale_crameri_vanimo`, `scale_crameri_vik0`, `scale_crameri_vik`

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

**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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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_roma0,
scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku, scale_crameri_vanimo,
scale_crameri_vik0, scale_crameri_vik
```

### 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\_roma0    *Fabio Crameri's romaO Cyclic Color Scheme*

---

### Description

Fabio Crameri's *romaO* Cyclic Color Scheme

### Usage

```
scale_colour_roma0(
  ...,
  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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> scalar: should the color scheme be used as a discrete scale?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

**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\\_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\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_Y10rBr`, `scale_tol_incandescent`, `scale_tol_iridescent`, `scale_tol_smoothrainbow`

Other Fabio Crameri's color schemes: `scale_crameri_acton`, `scale_crameri_bam0`, `scale_crameri_bamako`, `scale_crameri_bam`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `scale_crameri_berlin`, `scale_crameri_bilbao`, `scale_crameri_broc0`, `scale_crameri_broc`, `scale_crameri_buda`, `scale_crameri_bukavu`, `scale_crameri_cork0`, `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_roma0,
scale_crameri_roma, scale_crameri_tofino, scale_crameri_turku, scale_crameri_vanimo,
scale_crameri_vik0, scale_crameri_vik
```

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

## Value

A **continuous** scale.

## Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

### 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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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\\_Yl0rBr](#), [scale\\_tol\\_incandescent](#), [scale\\_tol\\_iridescent](#), [scale\\_tol\\_smoothrainbow](#)

Other Fabio Crameri's color schemes: [scale\\_crameri\\_acton](#), [scale\\_crameri\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

### 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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vik0](#), [scale\\_crameri\\_vik](#)

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

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale().#'`

**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\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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_roma0,
scale_crameri_roma, scale_crameri_tofino, scale_crameri_tokyo, scale_crameri_turku,
scale_crameri_vanimo, 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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> scalar: should the color scheme be used as a discrete scale?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_roma0](#)

Other Fabio Crameri's color schemes: [scale\\_crameri\\_acton](#), [scale\\_crameri\\_bam0](#), [scale\\_crameri\\_bamako](#), [scale\\_crameri\\_bam](#), [scale\\_crameri\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [scale\\_crameri\\_berlin](#), [scale\\_crameri\\_bilbao](#), [scale\\_crameri\\_broc0](#), [scale\\_crameri\\_broc](#), [scale\\_crameri\\_buda](#), [scale\\_crameri\\_bukavu](#), [scale\\_crameri\\_cork0](#), [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\\_roma0](#), [scale\\_crameri\\_roma](#), [scale\\_crameri\\_tofino](#), [scale\\_crameri\\_tokyo](#), [scale\\_crameri\\_turku](#), [scale\\_crameri\\_vanimo](#), [scale\\_crameri\\_vik](#)

## 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\_logical\_discrete

*Binary Discrete Color Schemes for **ggplot2** and **ggraph***

---

## Description

Provides a color scale for [logical](#) values.

## Usage

```
scale_colour_logical(..., aesthetics = "colour")

scale_color_logical(..., aesthetics = "colour")

scale_fill_logical(..., aesthetics = "fill")
```

## Arguments

...	Arguments passed to <code>ggplot2::discrete_scale()</code> .
aesthetics	A <a href="#">character</a> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

## Value

A [discrete](#) scale.

## Author(s)

N. Frerebeau

## 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](#), [scale\\_tol\\_vibrant](#)

---

`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 `ggplot2::discrete_scale()`.

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 color-blind safe color schemes: `scale_crameri_cyclic`, `scale_crameri_diverging`, `scale_crameri_mutliseque`, `scale_crameri_sequential`, `scale_tol_discrete`, `scale_tol_diverging`, `scale_tol_sequential`

Other qualitative color schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_logical_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)

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

## See Also

Other color palettes: [colour\(\)](#), [info\(\)](#), [ramp\(\)](#)

## 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 <code>ggplot2::discrete_scale()</code> .
reverse	A <a href="#">logical</a> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <a href="#">character</a> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

### Value

A [discrete](#) scale.

### Interpolation

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

### Qualitative Color Schemes

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 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 <a href="#">logical</a> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <a href="#">character</a> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A [discrete](#) scale.

## Interpolation

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

## Qualitative Color Schemes

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_logical_discrete`, `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 <code>ggplot2::continuous_scale()</code> .
reverse	A <b>logical</b> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

## Value

A **discrete** scale.

## Interpolation

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, with the exception of the discrete rainbow scheme (see below).

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

## 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_logical_discrete`, `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 <code>ggplot2::discrete_scale()</code> .
reverse	A <a href="#">logical</a> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <a href="#">character</a> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

## Value

A [discrete](#) scale.



## Interpolation

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

## Qualitative Color Schemes

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_logical_discrete`, `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_color"  
)
```

```

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

### Value

A **continuous** scale.

### Interpolation

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, with the exception of the discrete rainbow scheme (see below).

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 sequential color schemes: `scale_crameri_acton`, `scale_crameri_bamako`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> 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.

### Interpolation

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, with the exception of the discrete rainbow scheme (see below).

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 sequential color schemes: `scale_crameri_acton`, `scale_crameri_bamako`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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_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_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_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\_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.

### Interpolation

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

### Qualitative Color Schemes

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\\_logical\\_discrete](#), [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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A **discrete** scale.

**Interpolation**

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

**Qualitative Color Schemes**

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_logical_discrete`, `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_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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A **discrete** scale.

**Interpolation**

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

**Qualitative Color Schemes**

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_logical_discrete`, `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 `ggplot2::continuous_scale()`.

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

range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
midpoint	A length-one <b>numeric</b> vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A <b>logical</b> 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 <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 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 = 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\_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 <code>ggplot2::discrete_scale()</code> .
reverse	A <a href="#">logical</a> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <a href="#">character</a> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

## Value

A [discrete](#) scale.

## Interpolation

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

## Qualitative Color Schemes

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\\_logical\\_discrete](#), [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      *Paul Tol's PRGn Diverging Color Scheme*

---

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

### Value

A **continuous** scale.

### Interpolation

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 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\\_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\\_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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> 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 <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A [continuous](#) scale.

**Interpolation**

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, with the exception of the discrete rainbow scheme (see below).

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

**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\\_batlowK](#), [scale\\_crameri\\_batlowW](#), [scale\\_crameri\\_batlow](#), [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](#)

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

### Value

A **continuous** scale.

**Interpolation**

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.

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <b>character</b> string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

**Value**

A **discrete** scale.

**Interpolation**

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

**Qualitative Color Schemes**

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_logical_discrete`, `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 <b>logical</b> scalar. Should the resulting vector of colors be reversed?
range	A length-two <b>numeric</b> vector specifying the fraction of the scheme's color domain to keep.
discrete	A <b>logical</b> 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.

### Interpolation

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, with the exception of the discrete rainbow scheme (see below).

Note that the default color for NA can be overridden by passing a value to `ggplot2::continuous_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 sequential color schemes: `scale_crameri_acton`, `scale_crameri_bamako`, `scale_crameri_batlowK`, `scale_crameri_batlowW`, `scale_crameri_batlow`, `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, 21
- scale\_crameri\_bam, 23
- scale\_crameri\_bamako, 26
- scale\_crameri\_bam0, 29
- scale\_crameri\_batlow, 31
- scale\_crameri\_batlowK, 33
- scale\_crameri\_batlowW, 36
- scale\_crameri\_berlin, 39
- scale\_crameri\_bilbao, 41
- scale\_crameri\_broc, 44
- scale\_crameri\_broc0, 47
- scale\_crameri\_buda, 49
- scale\_crameri\_bukavu, 51
- scale\_crameri\_cork, 53
- scale\_crameri\_cork0, 56
- scale\_crameri\_davos, 58
- scale\_crameri\_devon, 60
- scale\_crameri\_fes, 63
- scale\_crameri\_grayC, 65
- scale\_crameri\_hawaii, 68
- scale\_crameri\_imola, 70
- scale\_crameri\_lajolla, 73
- scale\_crameri\_lapaz, 76
- scale\_crameri\_lisbon, 78
- scale\_crameri\_nuuk, 81
- scale\_crameri\_oleron, 84
- scale\_crameri\_oslo, 86
- scale\_crameri\_roma, 88
- scale\_crameri\_roma0, 91
- scale\_crameri\_tofino, 93
- scale\_crameri\_tokyo, 95
- scale\_crameri\_turku, 98
- scale\_crameri\_vanimo, 101
- scale\_crameri\_vik, 103
- scale\_crameri\_vik0, 106

## \* Okabe and Ito's color scheme

- scale\_okabeito\_discrete, 109

## \* Paul Tol's color schemes

- scale\_tol\_bright, 112
- scale\_tol\_BuRd, 113
- scale\_tol\_dark, 116
- scale\_tol\_discreterainbow, 118
- scale\_tol\_highcontrast, 120
- scale\_tol\_incandescent, 122
- scale\_tol\_iridescent, 124
- scale\_tol\_light, 127
- scale\_tol\_mediumcontrast, 128
- scale\_tol\_muted, 130
- scale\_tol\_nightfall, 132
- scale\_tol\_pale, 135
- scale\_tol\_PRGn, 137
- scale\_tol\_smoothrainbow, 139
- scale\_tol\_sunset, 142
- scale\_tol\_vibrant, 144
- scale\_tol\_YlOrBr, 146

## \* color palettes

- colour, 3
- info, 9
- ramp, 15
- scale\_picker, 111

## \* color-blind safe color schemes

- scale\_okabeito\_discrete, 109

## \* color

- colour, 3
- ramp, 15

## \* cyclic color schemes

- scale\_crameri\_bam0, 29
- scale\_crameri\_broc0, 47
- scale\_crameri\_cork0, 56
- scale\_crameri\_roma0, 91
- scale\_crameri\_vik0, 106

## \* diagnostic tools

- compare, 7
- convert, 8
- plot.color\_scheme, 10
- plot\_map, 11
- plot\_scheme, 12

- plot\_scheme\_colourblind, 13
- plot\_tiles, 14
- \* **diverging color schemes**
  - scale\_crameri\_bam, 23
  - scale\_crameri\_berlin, 39
  - scale\_crameri\_broc, 44
  - scale\_crameri\_cork, 53
  - scale\_crameri\_lisbon, 78
  - scale\_crameri\_roma, 88
  - scale\_crameri\_tofino, 93
  - scale\_crameri\_vanimo, 101
  - scale\_crameri\_vik, 103
  - scale\_tol\_BuRd, 113
  - scale\_tol\_nightfall, 132
  - scale\_tol\_PRGn, 137
  - scale\_tol\_sunset, 142
- \* **multi sequential color schemes**
  - scale\_crameri\_bukavu, 51
  - scale\_crameri\_fes, 63
  - scale\_crameri\_oleron, 84
- \* **qualitative color schemes**
  - scale\_colour\_land, 16
  - scale\_colour\_soil, 18
  - scale\_colour\_stratigraphy, 19
  - scale\_logical\_discrete, 108
  - scale\_okabeito\_discrete, 109
  - scale\_tol\_bright, 112
  - scale\_tol\_dark, 116
  - scale\_tol\_discreterainbow, 118
  - scale\_tol\_highcontrast, 120
  - scale\_tol\_light, 127
  - scale\_tol\_mediumcontrast, 128
  - scale\_tol\_muted, 130
  - scale\_tol\_pale, 135
  - scale\_tol\_vibrant, 144
- \* **sequential color schemes**
  - scale\_crameri\_acton, 21
  - scale\_crameri\_bamako, 26
  - scale\_crameri\_batlow, 31
  - scale\_crameri\_batlowK, 33
  - scale\_crameri\_batlowW, 36
  - scale\_crameri\_bilbao, 41
  - scale\_crameri\_buda, 49
  - scale\_crameri\_davos, 58
  - scale\_crameri\_devon, 60
  - scale\_crameri\_grayC, 65
  - scale\_crameri\_hawaii, 68
  - scale\_crameri\_imola, 70
  - scale\_crameri\_lajolla, 73
  - scale\_crameri\_lapaz, 76
  - scale\_crameri\_nuuk, 81
  - scale\_crameri\_oslo, 86
  - scale\_crameri\_tokyo, 95
  - scale\_crameri\_turku, 98
  - scale\_tol\_incandescent, 122
  - scale\_tol\_iridescent, 124
  - scale\_tol\_smoothrainbow, 139
  - scale\_tol\_YlOrBr, 146
- \* **themed color schemes**
  - scale\_colour\_land, 16
  - scale\_colour\_soil, 18
  - scale\_colour\_stratigraphy, 19
- character, 4, 7, 8, 10–18, 20, 22, 25, 27, 30, 32, 35, 37, 40, 43, 45, 47, 50, 52, 55, 57, 59, 62, 64, 66, 69, 72, 74, 77, 80, 82, 84, 87, 90, 92, 94, 97, 99, 102, 105, 107, 108, 110–112, 115, 116, 118, 120, 123, 126, 127, 129, 131, 134, 135, 138, 140, 143, 145, 148
- color (colour), 3
- color(), 8, 10, 15
- colorRampPalette, 4
- colour, 3, 10, 15, 111
- compare, 7, 9–14
- continuous, 22, 25, 28, 30, 32, 35, 37, 40, 43, 45, 47, 50, 52, 55, 57, 59, 62, 64, 66, 69, 72, 74, 77, 80, 82, 84, 87, 90, 92, 94, 97, 99, 102, 105, 107, 111, 115, 123, 126, 134, 138, 141, 143, 148
- convert, 7, 8, 10–14
- data.frame, 10
- discrete, 17, 18, 20, 108, 110–112, 116, 118, 120, 127, 129, 131, 135, 145
- distance matrix, 7
- function, 8, 15
- ggplot2::continuous\_scale(), 22, 25, 27–30, 32, 35, 37, 38, 40, 42, 43, 45–48, 50, 52, 55, 57, 59, 62, 64, 66, 67, 69, 71, 72, 74, 77, 80, 82, 84, 85, 87, 90, 92, 94, 97, 99, 100, 102, 105, 107, 115, 118, 119, 123, 125, 126, 133, 134, 138, 140, 141, 143, 144, 147, 148

- ggplot2::discrete\_scale(), [16](#), [18](#), [20](#),  
[108](#), [110](#), [112](#), [116](#), [120](#), [127](#), [129](#),  
[131](#), [135](#), [145](#)
- graphics::par(), [12](#)
- info, [6](#), [9](#), [15](#), [111](#)
- info(), [111](#)
- integer, [4](#), [7](#), [14](#)
- logical, [4](#), [7](#), [12](#), [22](#), [25](#), [27](#), [29](#), [30](#), [32](#), [35](#),  
[37](#), [40](#), [42](#), [43](#), [45](#), [47](#), [50](#), [52](#), [55](#), [57](#),  
[59](#), [62](#), [64](#), [66](#), [69](#), [71](#), [72](#), [74](#), [77](#), [80](#),  
[82](#), [84](#), [87](#), [90](#), [92](#), [94](#), [97](#), [99](#), [102](#),  
[105](#), [107](#), [108](#), [110](#), [112](#), [115](#), [116](#),  
[118](#), [120](#), [123](#), [125](#), [127](#), [129](#), [131](#),  
[133–135](#), [138](#), [140](#), [143](#), [145](#), [147](#)
- numeric, [12](#), [22](#), [25](#), [27](#), [29](#), [32](#), [35](#), [37](#), [40](#), [42](#),  
[45](#), [47](#), [50](#), [52](#), [55](#), [57](#), [59](#), [62](#), [64](#), [66](#),  
[69](#), [71](#), [74](#), [77](#), [80](#), [82](#), [84](#), [87](#), [90](#), [92](#),  
[94](#), [97](#), [99](#), [102](#), [105](#), [107](#), [115](#), [123](#),  
[125](#), [134](#), [138](#), [140](#), [143](#), [147](#)
- plot.color\_scheme, [7](#), [9](#), [10](#), [11–14](#)
- plot\_map, [7](#), [9](#), [10](#), [11](#), [12–14](#)
- plot\_scheme, [7](#), [9–11](#), [12](#), [13](#), [14](#)
- plot\_scheme\_colorblind  
(plot\_scheme\_colourblind), [13](#)
- plot\_scheme\_colourblind, [7](#), [9–12](#), [13](#), [14](#)
- plot\_tiles, [7](#), [9–13](#), [14](#)
- ramp, [6](#), [10](#), [15](#), [111](#)
- scale\_color\_acton  
(scale\_crameri\_acton), [21](#)
- scale\_color\_bam (scale\_crameri\_bam), [23](#)
- scale\_color\_bamako  
(scale\_crameri\_bamako), [26](#)
- scale\_color\_bam0 (scale\_crameri\_bam0),  
[29](#)
- scale\_color\_batlow  
(scale\_crameri\_batlow), [31](#)
- scale\_color\_batlowK  
(scale\_crameri\_batlowK), [33](#)
- scale\_color\_batlowW  
(scale\_crameri\_batlowW), [36](#)
- scale\_color\_berlin  
(scale\_crameri\_berlin), [39](#)
- scale\_color\_bilbao  
(scale\_crameri\_bilbao), [41](#)
- scale\_color\_bright (scale\_tol\_bright),  
[112](#)
- scale\_color\_broc (scale\_crameri\_broc),  
[44](#)
- scale\_color\_broc0  
(scale\_crameri\_broc0), [47](#)
- scale\_color\_buda (scale\_crameri\_buda),  
[49](#)
- scale\_color\_bukavu  
(scale\_crameri\_bukavu), [51](#)
- scale\_color\_BuRd (scale\_tol\_BuRd), [113](#)
- scale\_color\_cork (scale\_crameri\_cork),  
[53](#)
- scale\_color\_cork0  
(scale\_crameri\_cork0), [56](#)
- scale\_color\_dark (scale\_tol\_dark), [116](#)
- scale\_color\_davos  
(scale\_crameri\_davos), [58](#)
- scale\_color\_devon  
(scale\_crameri\_devon), [60](#)
- scale\_color\_discreterainbow  
(scale\_tol\_discreterainbow),  
[118](#)
- scale\_color\_fes (scale\_crameri\_fes), [63](#)
- scale\_color\_grayC  
(scale\_crameri\_grayC), [65](#)
- scale\_color\_hawaii  
(scale\_crameri\_hawaii), [68](#)
- scale\_color\_highcontrast  
(scale\_tol\_highcontrast), [120](#)
- scale\_color\_imola  
(scale\_crameri\_imola), [70](#)
- scale\_color\_incandescent  
(scale\_tol\_incandescent), [122](#)
- scale\_color\_iridescent  
(scale\_tol\_iridescent), [124](#)
- scale\_color\_lajolla  
(scale\_crameri\_lajolla), [73](#)
- scale\_color\_land (scale\_colour\_land), [16](#)
- scale\_color\_lapaz  
(scale\_crameri\_lapaz), [76](#)
- scale\_color\_light (scale\_tol\_light), [127](#)
- scale\_color\_lisbon  
(scale\_crameri\_lisbon), [78](#)
- scale\_color\_logical  
(scale\_logical\_discrete), [108](#)
- scale\_color\_mediumcontrast  
(scale\_tol\_mediumcontrast), [128](#)

- scale\_color\_muted (scale\_tol\_muted), 130
- scale\_color\_nightfall
  - (scale\_tol\_nightfall), 132
- scale\_color\_nuuk (scale\_crameri\_nuuk), 81
- scale\_color\_okabeito
  - (scale\_okabeito\_discrete), 109
- scale\_color\_oleron
  - (scale\_crameri\_oleron), 84
- scale\_color\_oslo (scale\_crameri\_oslo), 86
- scale\_color\_pale (scale\_tol\_pale), 135
- scale\_color\_picker (scale\_picker), 111
- scale\_color\_PRGn (scale\_tol\_PRGn), 137
- scale\_color\_roma (scale\_crameri\_roma), 88
- scale\_color\_roma0
  - (scale\_crameri\_roma0), 91
- scale\_color\_smoothrainbow
  - (scale\_tol\_smoothrainbow), 139
- scale\_color\_soil (scale\_colour\_soil), 18
- scale\_color\_stratigraphy
  - (scale\_colour\_stratigraphy), 19
- scale\_color\_sunset (scale\_tol\_sunset), 142
- scale\_color\_tofino
  - (scale\_crameri\_tofino), 93
- scale\_color\_tokyo
  - (scale\_crameri\_tokyo), 95
- scale\_color\_turku
  - (scale\_crameri\_turku), 98
- scale\_color\_vanimo
  - (scale\_crameri\_vanimo), 101
- scale\_color\_vibrant
  - (scale\_tol\_vibrant), 144
- scale\_color\_vik (scale\_crameri\_vik), 103
- scale\_color\_vik0 (scale\_crameri\_vik0), 106
- scale\_color\_YlOrBr (scale\_tol\_YlOrBr), 146
- scale\_colour\_acton
  - (scale\_crameri\_acton), 21
- scale\_colour\_bam (scale\_crameri\_bam), 23
- scale\_colour\_bamako
  - (scale\_crameri\_bamako), 26
- scale\_colour\_bam0 (scale\_crameri\_bam0), 29
- scale\_colour\_batlow
  - (scale\_crameri\_batlow), 31
- scale\_colour\_batlowK
  - (scale\_crameri\_batlowK), 33
- scale\_colour\_batlowW
  - (scale\_crameri\_batlowW), 36
- scale\_colour\_berlin
  - (scale\_crameri\_berlin), 39
- scale\_colour\_bilbao
  - (scale\_crameri\_bilbao), 41
- scale\_colour\_bright (scale\_tol\_bright), 112
- scale\_colour\_broc (scale\_crameri\_broc), 44
- scale\_colour\_broc0
  - (scale\_crameri\_broc0), 47
- scale\_colour\_buda (scale\_crameri\_buda), 49
- scale\_colour\_bukavu
  - (scale\_crameri\_bukavu), 51
- scale\_colour\_BuRd (scale\_tol\_BuRd), 113
- scale\_colour\_cork (scale\_crameri\_cork), 53
- scale\_colour\_cork0
  - (scale\_crameri\_cork0), 56
- scale\_colour\_dark (scale\_tol\_dark), 116
- scale\_colour\_davos
  - (scale\_crameri\_davos), 58
- scale\_colour\_devon
  - (scale\_crameri\_devon), 60
- scale\_colour\_discreterainbow
  - (scale\_tol\_discreterainbow), 118
- scale\_colour\_fes (scale\_crameri\_fes), 63
- scale\_colour\_grayC
  - (scale\_crameri\_grayC), 65
- scale\_colour\_hawaii
  - (scale\_crameri\_hawaii), 68
- scale\_colour\_highcontrast
  - (scale\_tol\_highcontrast), 120
- scale\_colour\_imola
  - (scale\_crameri\_imola), 70
- scale\_colour\_incandescent
  - (scale\_tol\_incandescent), 122
- scale\_colour\_iridescent
  - (scale\_tol\_iridescent), 124
- scale\_colour\_lajolla
  - (scale\_crameri\_lajolla), 73
- scale\_colour\_land, 16, 19, 20, 108, 110,



- 113, 117, 119, 121, 128, 130, 132, 136, 146*
- scale\_colour\_lapaz
  - (scale\_crameri\_lapaz), 76
- scale\_colour\_light (scale\_tol\_light), 127
- scale\_colour\_lisbon
  - (scale\_crameri\_lisbon), 78
- scale\_colour\_logical
  - (scale\_logical\_discrete), 108
- scale\_colour\_mediumcontrast
  - (scale\_tol\_mediumcontrast), 128
- scale\_colour\_muted (scale\_tol\_muted), 130
- scale\_colour\_nightfall
  - (scale\_tol\_nightfall), 132
- scale\_colour\_nuuk (scale\_crameri\_nuuk), 81
- scale\_colour\_okabeito
  - (scale\_okabeito\_discrete), 109
- scale\_colour\_oleron
  - (scale\_crameri\_oleron), 84
- scale\_colour\_oslo (scale\_crameri\_oslo), 86
- scale\_colour\_pale (scale\_tol\_pale), 135
- scale\_colour\_picker (scale\_picker), 111
- scale\_colour\_PRGn (scale\_tol\_PRGn), 137
- scale\_colour\_roma (scale\_crameri\_roma), 88
- scale\_colour\_roma0
  - (scale\_crameri\_roma0), 91
- scale\_colour\_smoothrainbow
  - (scale\_tol\_smoothrainbow), 139
- scale\_colour\_soil, *17, 18, 20, 108, 110, 113, 117, 119, 121, 128, 130, 132, 136, 146*
- scale\_colour\_stratigraphy, *17, 19, 19, 108, 110, 113, 117, 119, 121, 128, 130, 132, 136, 146*
- scale\_colour\_sunset (scale\_tol\_sunset), 142
- scale\_colour\_tofino
  - (scale\_crameri\_tofino), 93
- scale\_colour\_tokyo
  - (scale\_crameri\_tokyo), 95
- scale\_colour\_turku
  - (scale\_crameri\_turku), 98
- scale\_colour\_vanimo
  - (scale\_crameri\_vanimo), 101
- scale\_colour\_vibrant
  - (scale\_tol\_vibrant), 144
- scale\_colour\_vik (scale\_crameri\_vik), 103
- scale\_colour\_vik0 (scale\_crameri\_vik0), 106
- scale\_colour\_YlOrBr (scale\_tol\_YlOrBr), 146
- scale\_crameri\_acton, *21, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148*
- scale\_crameri\_bam, *23, 23, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 115, 134, 139, 144*
- scale\_crameri\_bamako, *23, 26, 26, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148*
- scale\_crameri\_bam0, *23, 26, 28, 29, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107*
- scale\_crameri\_batlow, *23, 26, 28, 30, 31, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148*
- scale\_crameri\_batlowK, *23, 26, 28, 30, 33, 33, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148*
- scale\_crameri\_batlowW, *23, 26, 28, 30, 33, 35, 36, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148*
- scale\_crameri\_berlin, *23, 26, 28, 30, 33, 35, 38, 39, 43, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 115, 134, 139, 144*

- scale\_crameri\_bilbao*, 23, 26, 28, 30, 33, 35, 38, 41, 41, 46, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_broc*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 44, 48, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 115, 134, 139, 144  
*scale\_crameri\_broc0*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 47, 51, 53, 55, 57, 60, 62, 64, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107  
*scale\_crameri\_buda*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 49, 53, 55, 57, 60, 62, 65, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_bukavu*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 51, 55, 57, 60, 62, 64, 65, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107  
*scale\_crameri\_cork*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 53, 57, 60, 62, 65, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 115, 134, 139, 144  
*scale\_crameri\_cork0*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 56, 60, 62, 65, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107  
*scale\_crameri\_cyclic*, 110  
*scale\_crameri\_davos*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 58, 62, 65, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_devon*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 60, 65, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_diverging*, 110  
*scale\_crameri\_fes*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 63, 67, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107  
*scale\_crameri\_grayC*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 65, 65, 70, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_hawaii*, 23, 26, 28, 30, 33, 35, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 65, 67, 68, 72, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_imola*, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 63, 65, 67, 70, 70, 75, 78, 80, 83, 85, 88, 90, 92, 95, 97, 98, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_lajolla*, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 63, 65, 67, 70, 72, 73, 78, 80, 83, 85, 88, 90, 92, 95, 97, 98, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_lisbon*, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 63, 65, 67, 70, 72, 75, 78, 78, 83, 85, 88, 90, 92, 95, 98, 100, 103, 105, 107, 115, 134, 139, 144  
*scale\_crameri\_lapaz*, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 62, 63, 65, 67, 70, 72, 75, 76, 80, 83, 85, 88, 90, 92, 95, 97, 98, 100, 103, 105, 107, 124, 126, 141, 148  
*scale\_crameri\_lisbon*, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 55, 57, 60, 63, 65, 67, 70, 72, 75, 78, 78, 83, 85, 88, 90, 92, 95, 98, 100, 103, 105, 107, 115, 134, 139, 144  
*scale\_crameri\_mutlisequential*, 110  
*scale\_crameri\_nuuk*, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 62, 63, 65, 67, 70, 72, 75, 78, 80, 81, 85, 88, 91, 92, 95, 97, 98, 100, 103, 106, 107, 124, 126, 141, 148  
*scale\_crameri\_oleron*, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 63–65, 67, 70, 72, 75, 78, 81, 83, 84, 88, 91, 92, 95, 98, 100, 103, 106, 107  
*scale\_crameri\_oslo*, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 62, 63, 65, 67, 70, 72, 75, 78, 81, 83, 85, 86, 91, 92, 95, 97, 98, 100, 103, 106, 107, 124, 126, 141, 148

- scale\_crameri\_roma, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 55–57, 60, 63, 65, 67, 70, 72, 75, 78, 80, 81, 83, 85, 88, 88, 92, 95, 98, 100, 103, 105–107, 115, 134, 139, 144
- scale\_crameri\_roma0, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 63, 65, 67, 70, 72, 75, 78, 81, 83, 85, 88, 91, 91, 95, 98, 100, 103, 106, 107
- scale\_crameri\_sequential, 110
- scale\_crameri\_tofino, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 55–57, 60, 63, 65, 67, 70, 72, 75, 78, 80, 81, 83, 85, 88, 90–92, 93, 98, 100, 103, 105–107, 115, 134, 139, 144
- scale\_crameri\_tokyo, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 62, 63, 65, 67, 70, 72, 75, 78, 81, 83, 85, 88, 91, 92, 95, 95, 100, 103, 106, 107, 124, 126, 141, 148
- scale\_crameri\_turku, 23, 26, 28, 30, 33, 35, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 62, 63, 65, 67, 70, 72, 75, 78, 81, 83, 85, 88, 91, 92, 95, 97, 98, 98, 103, 106, 107, 124, 126, 141, 148
- scale\_crameri\_vanimo, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 55–57, 60, 63, 65, 67, 70, 72, 75, 78, 80, 81, 83, 85, 88, 90–92, 95, 98, 100, 101, 105–107, 115, 134, 139, 144
- scale\_crameri\_vik, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 55–57, 60, 63, 65, 67, 70, 72, 75, 78, 80, 81, 83, 85, 88, 90–92, 95, 98, 100, 103, 103, 107, 115, 134, 139, 144
- scale\_crameri\_vik0, 23, 26, 28, 30, 33, 36, 38, 41, 43, 46, 48, 51, 53, 56, 57, 60, 63, 65, 67, 70, 72, 75, 78, 81, 83, 85, 88, 91, 92, 95, 98, 100, 103, 106, 106
- scale\_edge\_color\_acton (scale\_crameri\_acton), 21
- scale\_edge\_color\_bam (scale\_crameri\_bam), 23
- scale\_edge\_color\_bamako (scale\_crameri\_bamako), 26
- scale\_edge\_color\_batlow (scale\_crameri\_batlow), 31
- scale\_edge\_color\_batlowK (scale\_crameri\_batlowK), 33
- scale\_edge\_color\_batlowW (scale\_crameri\_batlowW), 36
- scale\_edge\_color\_berlin (scale\_crameri\_berlin), 39
- scale\_edge\_color\_bilbao (scale\_crameri\_bilbao), 41
- scale\_edge\_color\_bright (scale\_tol\_bright), 112
- scale\_edge\_color\_broc (scale\_crameri\_broc), 44
- scale\_edge\_color\_buda (scale\_crameri\_buda), 49
- scale\_edge\_color\_BuRd (scale\_tol\_BuRd), 113
- scale\_edge\_color\_cork (scale\_crameri\_cork), 53
- scale\_edge\_color\_dark (scale\_tol\_dark), 116
- scale\_edge\_color\_davos (scale\_crameri\_davos), 58
- scale\_edge\_color\_devon (scale\_crameri\_devon), 60
- scale\_edge\_color\_discreterainbow (scale\_tol\_discreterainbow), 118
- scale\_edge\_color\_grayC (scale\_crameri\_grayC), 65
- scale\_edge\_color\_hawaii (scale\_crameri\_hawaii), 68
- scale\_edge\_color\_highcontrast (scale\_tol\_highcontrast), 120
- scale\_edge\_color\_imola (scale\_crameri\_imola), 70
- scale\_edge\_color\_incandescent (scale\_tol\_incandescent), 122
- scale\_edge\_color\_iridescent (scale\_tol\_iridescent), 124
- scale\_edge\_color\_lajolla (scale\_crameri\_lajolla), 73
- scale\_edge\_color\_land (scale\_colour\_land), 16
- scale\_edge\_color\_lapaz (scale\_crameri\_lapaz), 76
- scale\_edge\_color\_light (scale\_tol\_light), 127
- scale\_edge\_color\_lisbon (scale\_crameri\_lisbon), 78

- scale\_edge\_color\_mediumcontrast  
(scale\_tol\_mediumcontrast), 128
- scale\_edge\_color\_muted  
(scale\_tol\_muted), 130
- scale\_edge\_color\_nightfall  
(scale\_tol\_nightfall), 132
- scale\_edge\_color\_nuuk  
(scale\_crameri\_nuuk), 81
- scale\_edge\_color\_okabeito  
(scale\_okabeito\_discrete), 109
- scale\_edge\_color\_oslo  
(scale\_crameri\_oslo), 86
- scale\_edge\_color\_pale (scale\_tol\_pale),  
135
- scale\_edge\_color\_picker (scale\_picker),  
111
- scale\_edge\_color\_PRGn (scale\_tol\_PRGn),  
137
- scale\_edge\_color\_roma  
(scale\_crameri\_roma), 88
- scale\_edge\_color\_smoothrainbow  
(scale\_tol\_smoothrainbow), 139
- scale\_edge\_color\_soil  
(scale\_colour\_soil), 18
- scale\_edge\_color\_stratigraphy  
(scale\_colour\_stratigraphy), 19
- scale\_edge\_color\_sunset  
(scale\_tol\_sunset), 142
- scale\_edge\_color\_tofino  
(scale\_crameri\_tofino), 93
- scale\_edge\_color\_tokyo  
(scale\_crameri\_tokyo), 95
- scale\_edge\_color\_turku  
(scale\_crameri\_turku), 98
- scale\_edge\_color\_vanimo  
(scale\_crameri\_vanimo), 101
- scale\_edge\_color\_vibrant  
(scale\_tol\_vibrant), 144
- scale\_edge\_color\_vik  
(scale\_crameri\_vik), 103
- scale\_edge\_color\_YlOrBr  
(scale\_tol\_YlOrBr), 146
- scale\_edge\_colour\_acton  
(scale\_crameri\_acton), 21
- scale\_edge\_colour\_bam  
(scale\_crameri\_bam), 23
- scale\_edge\_colour\_bamako  
(scale\_crameri\_bamako), 26
- scale\_edge\_colour\_batlow  
(scale\_crameri\_batlow), 31
- scale\_edge\_colour\_batlowK  
(scale\_crameri\_batlowK), 33
- scale\_edge\_colour\_batlowW  
(scale\_crameri\_batlowW), 36
- scale\_edge\_colour\_berlin  
(scale\_crameri\_berlin), 39
- scale\_edge\_colour\_bilbao  
(scale\_crameri\_bilbao), 41
- scale\_edge\_colour\_bright  
(scale\_tol\_bright), 112
- scale\_edge\_colour\_broc  
(scale\_crameri\_broc), 44
- scale\_edge\_colour\_buda  
(scale\_crameri\_buda), 49
- scale\_edge\_colour\_BuRd  
(scale\_tol\_BuRd), 113
- scale\_edge\_colour\_cork  
(scale\_crameri\_cork), 53
- scale\_edge\_colour\_dark  
(scale\_tol\_dark), 116
- scale\_edge\_colour\_davos  
(scale\_crameri\_davos), 58
- scale\_edge\_colour\_devon  
(scale\_crameri\_devon), 60
- scale\_edge\_colour\_discreterainbow  
(scale\_tol\_discreterainbow),  
118
- scale\_edge\_colour\_grayC  
(scale\_crameri\_grayC), 65
- scale\_edge\_colour\_hawaii  
(scale\_crameri\_hawaii), 68
- scale\_edge\_colour\_highcontrast  
(scale\_tol\_highcontrast), 120
- scale\_edge\_colour\_imola  
(scale\_crameri\_imola), 70
- scale\_edge\_colour\_incandescent  
(scale\_tol\_incandescent), 122
- scale\_edge\_colour\_iridescent  
(scale\_tol\_iridescent), 124
- scale\_edge\_colour\_lajolla  
(scale\_crameri\_lajolla), 73
- scale\_edge\_colour\_land  
(scale\_colour\_land), 16
- scale\_edge\_colour\_lapaz  
(scale\_crameri\_lapaz), 76
- scale\_edge\_colour\_light

- (scale\_tol\_light), 127
- scale\_edge\_colour\_lisbon
  - (scale\_crameri\_lisbon), 78
- scale\_edge\_colour\_mediumcontrast
  - (scale\_tol\_mediumcontrast), 128
- scale\_edge\_colour\_muted
  - (scale\_tol\_muted), 130
- scale\_edge\_colour\_nightfall
  - (scale\_tol\_nightfall), 132
- scale\_edge\_colour\_nuuk
  - (scale\_crameri\_nuuk), 81
- scale\_edge\_colour\_okabeito
  - (scale\_okabeito\_discrete), 109
- scale\_edge\_colour\_oslo
  - (scale\_crameri\_oslo), 86
- scale\_edge\_colour\_pale
  - (scale\_tol\_pale), 135
- scale\_edge\_colour\_picker
  - (scale\_picker), 111
- scale\_edge\_colour\_PRGn
  - (scale\_tol\_PRGn), 137
- scale\_edge\_colour\_roma
  - (scale\_crameri\_roma), 88
- scale\_edge\_colour\_smoothrainbow
  - (scale\_tol\_smoothrainbow), 139
- scale\_edge\_colour\_soil
  - (scale\_colour\_soil), 18
- scale\_edge\_colour\_stratigraphy
  - (scale\_colour\_stratigraphy), 19
- scale\_edge\_colour\_sunset
  - (scale\_tol\_sunset), 142
- scale\_edge\_colour\_tofino
  - (scale\_crameri\_tofino), 93
- scale\_edge\_colour\_tokyo
  - (scale\_crameri\_tokyo), 95
- scale\_edge\_colour\_turku
  - (scale\_crameri\_turku), 98
- scale\_edge\_colour\_vanimo
  - (scale\_crameri\_vanimo), 101
- scale\_edge\_colour\_vibrant
  - (scale\_tol\_vibrant), 144
- scale\_edge\_colour\_vik
  - (scale\_crameri\_vik), 103
- scale\_edge\_colour\_Y10rBr
  - (scale\_tol\_Y10rBr), 146
- scale\_edge\_fill\_acton
  - (scale\_crameri\_acton), 21
- scale\_edge\_fill\_bam
  - (scale\_crameri\_bam), 23
- scale\_edge\_fill\_bamako
  - (scale\_crameri\_bamako), 26
- scale\_edge\_fill\_batlow
  - (scale\_crameri\_batlow), 31
- scale\_edge\_fill\_batlowK
  - (scale\_crameri\_batlowK), 33
- scale\_edge\_fill\_batlowW
  - (scale\_crameri\_batlowW), 36
- scale\_edge\_fill\_berlin
  - (scale\_crameri\_berlin), 39
- scale\_edge\_fill\_bilbao
  - (scale\_crameri\_bilbao), 41
- scale\_edge\_fill\_bright
  - (scale\_tol\_bright), 112
- scale\_edge\_fill\_broc
  - (scale\_crameri\_broc), 44
- scale\_edge\_fill\_buda
  - (scale\_crameri\_buda), 49
- scale\_edge\_fill\_BuRd (scale\_tol\_BuRd),
  - 113
- scale\_edge\_fill\_cork
  - (scale\_crameri\_cork), 53
- scale\_edge\_fill\_dark (scale\_tol\_dark),
  - 116
- scale\_edge\_fill\_davos
  - (scale\_crameri\_davos), 58
- scale\_edge\_fill\_devon
  - (scale\_crameri\_devon), 60
- scale\_edge\_fill\_discreterainbow
  - (scale\_tol\_discreterainbow),
    - 118
- scale\_edge\_fill\_grayC
  - (scale\_crameri\_grayC), 65
- scale\_edge\_fill\_hawaii
  - (scale\_crameri\_hawaii), 68
- scale\_edge\_fill\_highcontrast
  - (scale\_tol\_highcontrast), 120
- scale\_edge\_fill\_imola
  - (scale\_crameri\_imola), 70
- scale\_edge\_fill\_incandescent
  - (scale\_tol\_incandescent), 122
- scale\_edge\_fill\_iridescent
  - (scale\_tol\_iridescent), 124
- scale\_edge\_fill\_lajolla
  - (scale\_crameri\_lajolla), 73
- scale\_edge\_fill\_land
  - (scale\_colour\_land), 16

- scale\_edge\_fill\_lapaz  
(scale\_crameri\_lapaz), 76
- scale\_edge\_fill\_light  
(scale\_tol\_light), 127
- scale\_edge\_fill\_lisbon  
(scale\_crameri\_lisbon), 78
- scale\_edge\_fill\_mediumcontrast  
(scale\_tol\_mediumcontrast), 128
- scale\_edge\_fill\_muted  
(scale\_tol\_muted), 130
- scale\_edge\_fill\_nightfall  
(scale\_tol\_nightfall), 132
- scale\_edge\_fill\_nuuk  
(scale\_crameri\_nuuk), 81
- scale\_edge\_fill\_okabeito  
(scale\_okabeito\_discrete), 109
- scale\_edge\_fill\_oslo  
(scale\_crameri\_oslo), 86
- scale\_edge\_fill\_pale (scale\_tol\_pale),  
135
- scale\_edge\_fill\_picker (scale\_picker),  
111
- scale\_edge\_fill\_PRGn (scale\_tol\_PRGn),  
137
- scale\_edge\_fill\_roma  
(scale\_crameri\_roma), 88
- scale\_edge\_fill\_smoothrainbow  
(scale\_tol\_smoothrainbow), 139
- scale\_edge\_fill\_soil  
(scale\_colour\_soil), 18
- scale\_edge\_fill\_stratigraphy  
(scale\_colour\_stratigraphy), 19
- scale\_edge\_fill\_sunset  
(scale\_tol\_sunset), 142
- scale\_edge\_fill\_tofino  
(scale\_crameri\_tofino), 93
- scale\_edge\_fill\_tokyo  
(scale\_crameri\_tokyo), 95
- scale\_edge\_fill\_turku  
(scale\_crameri\_turku), 98
- scale\_edge\_fill\_vanimo  
(scale\_crameri\_vanimo), 101
- scale\_edge\_fill\_vibrant  
(scale\_tol\_vibrant), 144
- scale\_edge\_fill\_vik  
(scale\_crameri\_vik), 103
- scale\_edge\_fill\_YlOrBr  
(scale\_tol\_YlOrBr), 146
- scale\_fill\_acton (scale\_crameri\_acton),  
21
- scale\_fill\_bam (scale\_crameri\_bam), 23
- scale\_fill\_bamako  
(scale\_crameri\_bamako), 26
- scale\_fill\_bam0 (scale\_crameri\_bam0), 29
- scale\_fill\_batlow  
(scale\_crameri\_batlow), 31
- scale\_fill\_batlowK  
(scale\_crameri\_batlowK), 33
- scale\_fill\_batlowW  
(scale\_crameri\_batlowW), 36
- scale\_fill\_berlin  
(scale\_crameri\_berlin), 39
- scale\_fill\_bilbao  
(scale\_crameri\_bilbao), 41
- scale\_fill\_bright (scale\_tol\_bright),  
112
- scale\_fill\_broc (scale\_crameri\_broc), 44
- scale\_fill\_broc0 (scale\_crameri\_broc0),  
47
- scale\_fill\_buda (scale\_crameri\_buda), 49
- scale\_fill\_bukavu  
(scale\_crameri\_bukavu), 51
- scale\_fill\_BuRd (scale\_tol\_BuRd), 113
- scale\_fill\_cork (scale\_crameri\_cork), 53
- scale\_fill\_cork0 (scale\_crameri\_cork0),  
56
- scale\_fill\_dark (scale\_tol\_dark), 116
- scale\_fill\_davos (scale\_crameri\_davos),  
58
- scale\_fill\_devon (scale\_crameri\_devon),  
60
- scale\_fill\_discreterainbow  
(scale\_tol\_discreterainbow),  
118
- scale\_fill\_fes (scale\_crameri\_fes), 63
- scale\_fill\_grayC (scale\_crameri\_grayC),  
65
- scale\_fill\_hawaii  
(scale\_crameri\_hawaii), 68
- scale\_fill\_highcontrast  
(scale\_tol\_highcontrast), 120
- scale\_fill\_imola (scale\_crameri\_imola),  
70
- scale\_fill\_incandescent  
(scale\_tol\_incandescent), 122
- scale\_fill\_iridescent

- (scale\_tol\_iridescent), 124
- scale\_fill\_lajolla
  - (scale\_crameri\_lajolla), 73
- scale\_fill\_land(scale\_colour\_land), 16
- scale\_fill\_lapaz(scale\_crameri\_lapaz), 76
- scale\_fill\_light(scale\_tol\_light), 127
- scale\_fill\_lisbon
  - (scale\_crameri\_lisbon), 78
- scale\_fill\_logical
  - (scale\_logical\_discrete), 108
- scale\_fill\_mediumcontrast
  - (scale\_tol\_mediumcontrast), 128
- scale\_fill\_nightfall
  - (scale\_tol\_nightfall), 132
- scale\_fill\_nuuk(scale\_crameri\_nuuk), 81
- scale\_fill\_okabeito
  - (scale\_okabeito\_discrete), 109
- scale\_fill\_oleron
  - (scale\_crameri\_oleron), 84
- scale\_fill\_oslo(scale\_crameri\_oslo), 86
- scale\_fill\_pale(scale\_tol\_pale), 135
- scale\_fill\_picker(scale\_picker), 111
- scale\_fill\_PRGn(scale\_tol\_PRGn), 137
- scale\_fill\_roma(scale\_crameri\_roma), 88
- scale\_fill\_roma0(scale\_crameri\_roma0), 91
- scale\_fill\_smoothrainbow
  - (scale\_tol\_smoothrainbow), 139
- scale\_fill\_soil(scale\_colour\_soil), 18
- scale\_fill\_stratigraphy
  - (scale\_colour\_stratigraphy), 19
- scale\_fill\_sunset(scale\_tol\_sunset), 142
- scale\_fill\_tofino
  - (scale\_crameri\_tofino), 93
- scale\_fill\_tokyo(scale\_crameri\_tokyo), 95
- scale\_fill\_turku(scale\_crameri\_turku), 98
- scale\_fill\_vanimo
  - (scale\_crameri\_vanimo), 101
- scale\_fill\_vibrant(scale\_tol\_vibrant), 144
- scale\_fill\_vik(scale\_crameri\_vik), 103
- scale\_fill\_vik0(scale\_crameri\_vik0), 106
- scale\_fill\_YlOrBr(scale\_tol\_YlOrBr), 146
- scale\_logical\_discrete, 17, 19, 20, 108, 110, 113, 117, 119, 121, 128, 130, 132, 136, 146
- scale\_okabeito\_discrete, 17, 19, 20, 108, 109, 113, 117, 119, 121, 128, 130, 132, 136, 146
- scale\_picker, 6, 10, 15, 111
- scale\_tol\_bright, 17, 19, 20, 108, 110, 112, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_BuRd, 26, 41, 46, 55, 80, 90, 95, 103, 105, 113, 113, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_dark, 17, 19, 20, 108, 110, 113, 115, 116, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_discrete, 110
- scale\_tol\_discreterainbow, 17, 19, 20, 108, 110, 113, 115, 117, 118, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_diverging, 110
- scale\_tol\_highcontrast, 17, 19, 20, 108, 110, 113, 115, 117, 119, 120, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_incandescent, 23, 28, 33, 35, 38, 43, 51, 60, 62, 67, 70, 72, 75, 78, 83, 88, 97, 100, 113, 115, 117, 119, 121, 122, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_iridescent, 23, 28, 33, 35, 38, 43, 51, 60, 62, 67, 70, 72, 75, 78, 83, 88, 97, 100, 113, 115, 117, 119, 121, 124, 124, 128, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_light, 17, 19, 20, 108, 110, 113, 115, 117, 119, 121, 124, 126, 127, 130, 132, 134, 136, 139, 141, 144, 146, 148
- scale\_tol\_mediumcontrast, 17, 19, 20, 108, 110, 113, 115, 117, 119, 121, 124, 126, 128, 128, 132, 134, 136, 139, 141, 144, 146, 148



`scale_tol_muted`, *17, 19, 20, 108, 110, 113, 115, 117, 119, 121, 124, 126, 128, 130, 130, 134, 136, 139, 141, 144, 146, 148*

`scale_tol_nightfall`, *26, 41, 46, 55, 80, 90, 95, 103, 105, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 132, 136, 139, 141, 144, 146, 148*

`scale_tol_pale`, *17, 19, 20, 108, 110, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 135, 139, 141, 144, 146, 148*

`scale_tol_PRGn`, *26, 41, 46, 55, 80, 90, 95, 103, 105, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 137, 141, 144, 146, 148*

`scale_tol_sequential`, *110*

`scale_tol_smoothrainbow`, *23, 28, 33, 35, 38, 43, 51, 60, 62, 67, 70, 72, 75, 78, 83, 88, 97, 100, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 139, 144, 146, 148*

`scale_tol_sunset`, *26, 41, 46, 55, 80, 90, 95, 103, 105, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 142, 146, 148*

`scale_tol_vibrant`, *17, 19, 20, 108, 110, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 144, 148*

`scale_tol_YlOrBr`, *23, 28, 33, 35, 38, 43, 51, 60, 62, 67, 70, 72, 75, 78, 83, 88, 97, 100, 113, 115, 117, 119, 121, 124, 126, 128, 130, 132, 134, 136, 139, 141, 144, 146, 146*

`spacesXYZ::DeltaE()`, *7*