

Package ‘khroma’

October 13, 2022

Title Colour Schemes for Scientific Data Visualization

Version 1.9.0

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

Description Colour schemes ready for each type of data (qualitative, diverging or sequential), with colours that are distinct for all people, including colour-blind readers. This package provides an implementation of Paul Tol (2018) and Fabio Crameri (2018) <[doi:10.5194/gmd-11-2541-2018](https://doi.org/10.5194/gmd-11-2541-2018)> colour schemes for use with 'graphics' or 'ggplot2'. It provides tools to simulate colour-blindness and to test how well the colours 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 (>= 3.3)

Imports graphics, grDevices, grid, stats, utils

Suggests covr, crayon, fansi, ggplot2, ggraph, knitr, rmarkdown, scales, spacesXYZ, testthat (>= 3.0.0), vdiff (>= 1.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.0

Collate 'anomalize.R' 'colour.R' 'compare.R' 'convert.R' 'ggplot2.R'
'khroma-package.R' 'plot.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)

Repository CRAN

Date/Publication 2022-06-18 13:50:02 UTC

R topics documented:

colour	3
compare	6
convert	8
info	9
plot	10
ramp	11
scale_colour_land	12
scale_colour_soil	14
scale_colour_stratigraphy	15
scale_crameri_acton	17
scale_crameri_bam	20
scale_crameri_bamako	22
scale_crameri_bamO	25
scale_crameri_batlow	27
scale_crameri_batlowK	30
scale_crameri_batlowW	32
scale_crameri_berlin	35
scale_crameri_bilbao	38
scale_crameri_broc	40
scale_crameri_brocO	43
scale_crameri_buda	45
scale_crameri_bukavu	47
scale_crameri_cork	49
scale_crameri_corkO	52
scale_crameri_davos	54
scale_crameri_devon	57
scale_crameri_fes	59
scale_crameri_grayC	61
scale_crameri_hawaii	64
scale_crameri_imola	67
scale_crameri_lajolla	69
scale_crameri_lapaz	72
scale_crameri_lisbon	75
scale_crameri_nuuk	77
scale_crameri_oleron	80
scale_crameri_oslo	82

scale_crameri_roma	85
scale_crameri_romaO	87
scale_crameri_tofino	89
scale_crameri_tokyo	92
scale_crameri_turku	95
scale_crameri_vanimo	97
scale_crameri_vik	100
scale_crameri_vikO	103
scale_logical_discrete	105
scale_okabeito_discrete	106
scale_picker	107
scale_tol_bright	108
scale_tol_BuRd	110
scale_tol_dark	112
scale_tol_discreterainbow	114
scale_tol_highcontrast	116
scale_tol_iridescent	118
scale_tol_light	121
scale_tol_mediumcontrast	123
scale_tol_muted	125
scale_tol_pale	126
scale_tol_PRGn	128
scale_tol_smoothrainbow	131
scale_tol_sunset	134
scale_tol_vibrant	136
scale_tol_YlOrBr	138

Index	142
--------------	------------

colour	<i>Colour Palette</i>
--------	-----------------------

Description

Provides qualitative, diverging and sequential colour 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 character string giving the name of the palette to be used (see below).
reverse	A logical scalar: should the resulting vector of colours should be reversed?
names	A logical scalar: should the names of the colours should be kept in the resulting vector?

lang	A character string specifying the language for the colour names. It must be one of "en" (English, the default) or "fr" (French).
force	A logical scalar. If TRUE, forces the colour scheme to be interpolated. It should not be used routinely with qualitative colour schemes, as they are designed to be used as is to remain colour-blind safe.
...	Further arguments passed to colorRampPalette .

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 colours.

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 colour that should be used for NA values.

max An **integer** giving the maximum number of colour values. Only relevant for non-interpolated colour schemes.

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

Paul Tol's Colour Schemes

The following palettes are available. The maximum number of supported colours is in brackets, this value is only relevant for the qualitative colour 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), BuRd (9), PRGn (9).

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

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour 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.
- Colour-blind people have difficulty distinguishing some colours of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two colour schemes that are reasonably clear in colour-blind vision. To remain colour-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 Colour Scheme

The following (qualitative) colour scheme is available:

`okabeito` Up to 8 colours.

Scientific Colour Schemes

The following (qualitative) color schemes are available:

`stratigraphy` International Chronostratigraphic Chart (175 colours).

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

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

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 colour 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

Colour Difference

Description

Computes CIELAB distance metric.

Usage

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

Arguments

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

Value

A [distance matrix](#).

Author(s)

N. Frerebeau

See Also

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

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 Colour-Blindness*

Description

Simulate Colour-Blindness

Usage

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

Arguments

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

Value

A palette [function](#) that returns a vector of anomalized colours. All the attributes of the initial palette function are inherited, with a supplementary attribute "mode" giving the corresponding colour-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](https://doi.org/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](https://doi.org/10.1002/(SICI)15206378(199908)24:4<243::AIDCOL5>3.0.CO;23).

See Also

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

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))
```

info

Information

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 colours 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 colour palettes: [colour\(\)](#), [ramp\(\)](#), [scale_picker](#)

Examples

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

plot

Plot Color Scheme

Description

- `plot()` allows to quickly display a color scheme returned by `colour()`.
- `plot_scheme()` shows colors in a plot.
- `plot_map()` and `plot_tiles()` produce a diagnostic map for a given color scheme.
- `plot_scheme_colorblind()` shows colors in a plot with different types of simulated color blindness.

Usage

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

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

plot_map(x)

plot_tiles(x, n = 512)

plot_scheme_colourblind(x)

plot_scheme_colorblind(x)
```

Arguments

<code>x</code>	A character vector of colors.
<code>...</code>	Currently not used.
<code>colours</code>	A logical scalar: should the hexadecimal representation of the colors be displayed?
<code>names</code>	A logical scalar: should the name of the colors be displayed?
<code>size</code>	A numeric value giving the amount by which plotting text should be magnified relative to the default. Works the same as <code>cex</code> parameter of <code>graphics::par()</code> .
<code>n</code>	An integer specifying the size of the grid (defaults to 512).

Author(s)

N. Frerebeau, V. Arel-Bundock

See Also

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

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 simulated color blindness
plot_scheme_colorblind(colour("bright")(7))
```

ramp

Colour Ramp

Description

Provides a colour scheme that map a numeric vector to colours.

Usage

```
ramp(x, palette, midpoint = NULL, ...)
```

Arguments

x	A numeric vector to be mapped to colours.
palette	A character string giving the name of the palette to be used (see colour()).
midpoint	A numeric value specifying the midpoint of x. If not NULL, x will be rescaled to have the specified midpoint.
...	Further arguments passed to colour() .

Value

A palette function with the following attributes, that when called with a single argument (the number of levels, e.g. the breaks returned by `hist()`) returns a vector of colours.

Author(s)

N. Frerebeau

See Also

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

Examples

```
## Visualize a simple DEM model

## Distribution of elevation values
elevation <- hist(volcano)

## Where are breaks?
elevation$breaks

## No rescaling
ramp_default <- ramp(elevation$breaks, palette = "BuRd")(10)
image(volcano, col = ramp_default)
legend("topright", legend = attr(ramp_default, "breaks"), fill = ramp_default)

## Rescale to midpoint
ramp_160 <- ramp(elevation$breaks, palette = "BuRd", midpoint = 160)(10)
image(volcano, col = ramp_160)
legend("topright", legend = attr(ramp_160, "breaks"), fill = ramp_160)
```

scale_colour_land	<i>AVHRR Global Land Cover Classification Colour Scheme for ggplot2 and ggraph</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 <code>character</code> 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 <code>character</code> 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 colour schemes: `scale_colour_soil()`, `scale_colour_stratigraphy()`

Other qualitative colour schemes: `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

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 colour scheme.

Usage

```

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

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

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

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

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

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

```

Arguments

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

Details

Values will be matched based on the soil names.

Value

A [discrete](#) scale.

Author(s)

N. Frerebeau

References

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

See Also

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

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

Examples

```
library(ggplot2)

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

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

scale_colour_stratigraphy

*Geologic Timescale Color Scheme for **ggplot2** and **ggraph***

Description

Provides the geologic timescale color scheme.

Usage

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

Arguments

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

Details

Values will be matched based on the geological unit names.

Value

A [discrete](#) scale.

Author(s)

N. Frerebeau

References

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

See Also

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

Other qualitative colour schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [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](#)

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 Colour Scheme*

Description

Fabio Crameri's *acton* Sequential Colour Scheme

Usage

```

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

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

```

```

    discrete = FALSE,
    aesthetics = "colour"
  )

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

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

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

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

```

Arguments

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

Value

A [continuous](#) scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *bam* Diverging Colour Scheme

Usage

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

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

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

```
scale_edge_colour_bam(  
  ...,  
  reverse = FALSE,
```

```

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *bamako* Sequential Colour Scheme

Usage

```
scale_colour_bamako(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_bamako(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_bamako(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_bamako(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_bamako(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_fill_bamako(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_fill"  
)
```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow`, `scale_tol_iridescent`, `scale_tol_smoothrainbow`

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *bamO* Cyclic Colour Scheme

Usage

```
scale_colour_bam0(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_bam0(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_bam0(  
  ...,
```

```

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: [scale_crameri_broc0](#), [scale_crameri_cork0](#), [scale_crameri_roma0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *batlow* Sequential Colour Scheme

Usage

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

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

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

```

    discrete = FALSE,
    aesthetics = "fill"
  )

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

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

Author(s)

N. Frerebeau

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

Other sequential colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *batlowK* Sequential Colour Scheme

Usage

```
scale_colour_batlowK(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_batlowK(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_batlowK(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_batlowK(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_batlowK(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"
```

```

)

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *batlowW* Sequential Colour Scheme

Usage

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



```
  aesthetics = "colour"
)

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

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

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

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

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

Arguments

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

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

Value

A [continuous](#) scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *berlin* Diverging Colour Scheme

Usage

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

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

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

```

    discrete = FALSE,
    aesthetics = "fill"
  )

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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 = 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_bilbao *Fabio Crameri's bilbao Sequential Colour Scheme*

Description

Fabio Crameri's *bilbao* Sequential Colour Scheme

Usage

```
scale_colour_bilbao(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_bilbao(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_bilbao(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_bilbao(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_bilbao(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"
```

```

)

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *broc* Diverging Colour Scheme

Usage

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



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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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 = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

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

```

scale_crameri_brocO *Fabio Crameri's brocO Cyclic Colour Scheme*

Description

Fabio Crameri's *brocO* Cyclic Colour Scheme

Usage

```

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: `scale_crameri_bam0`, `scale_crameri_cork0`, `scale_crameri_roma0`, `scale_crameri_vik0`

Other Fabio Crameri's colour 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 = unemploy)) +
  ggplot2::geom_point() +
  scale_colour_broc(reverse = TRUE, midpoint = 12000)

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

scale_crameri_buda *Fabio Crameri's buda Sequential Colour Scheme*

Description

Fabio Crameri's *buda* Sequential Colour Scheme

Usage

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

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

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

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

```

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *bukavu* Multi-Sequential Colour 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 logical scalar. Should the resulting vector of colours be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A **continuous** scale.

Interpolation

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

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

Author(s)

N. Frerebeau

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

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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_cork0 *Fabio Crameri's corkO Cyclic Colour Scheme*

Description

Fabio Crameri's *corkO* Cyclic Colour 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_corkO(
  ...,
  reverse = FALSE,
  range = c(0, 1),
  discrete = FALSE,
  aesthetics = "fill"
)

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: [scale_crameri_bam0](#), [scale_crameri_broc0](#), [scale_crameri_roma0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *davos* Sequential Colour Scheme

Usage

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

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

```
  aesthetics = "colour"
)

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

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

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

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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *devon* Sequential Colour 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_color"  
)
```

```

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme

Description

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

Usage

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

```

  aesthetics = "colour"
)

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: [scale_crameri_bukavu](#), [scale_crameri_oleron](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *grayC* Sequential Colour Scheme

Usage

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

```
  aesthetics = "colour"
)

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

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

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

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

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

Arguments

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

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

Value

A [continuous](#) scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *hawaii* Sequential Colour Scheme

Usage

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

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

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



```

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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_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_imola *Fabio Crameri's imola Sequential Colour Scheme*

Description

Fabio Crameri's *imola* Sequential Colour Scheme

Usage

```
scale_colour_imola(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_imola(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_imola(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_imola(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_imola(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"
```

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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *lajolla* Sequential Colour Scheme

Usage

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

```
  aesthetics = "colour"
)

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

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

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

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

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

Arguments

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

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

Value

A [continuous](#) scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *lapaz* Sequential Colour Scheme

Usage

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

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

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



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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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_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_lisbon *Fabio Crameri's lisbon Diverging Colour Scheme*

Description

Fabio Crameri's *lisbon* Diverging Colour Scheme

Usage

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

```

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *nuuk* Sequential Colour Scheme

Usage

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

```
  aesthetics = "colour"
)

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

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

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

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

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

Arguments

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

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

Value

A [continuous](#) scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

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

Usage

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

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

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


Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: [scale_crameri_bukavu](#), [scale_crameri_fes](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *oslo* Sequential Colour Scheme

Usage

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

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

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

scale_edge_colour_oslo(
```

```

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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_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_roma *Fabio Crameri's roma Diverging Colour Scheme*

Description

Fabio Crameri's *roma* Diverging Colour Scheme

Usage

```
scale_colour_roma(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_roma(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_roma(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_roma(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  midpoint = 0,  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_roma(  
  ...,
```

```

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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_romaO *Fabio Crameri's romaO Cyclic Colour Scheme*

Description

Fabio Crameri's *romaO* Cyclic Colour Scheme

Usage

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

```

  aesthetics = "colour"
)

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: [scale_crameri_bam0](#), [scale_crameri_broc0](#), [scale_crameri_cork0](#), [scale_crameri_vik0](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *tofino* Diverging Colour Scheme

Usage

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

```
    aesthetics = "colour"
  )

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

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

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

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

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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *tokyo* Sequential Colour Scheme

Usage

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

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

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

```

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_YlOrBr](#), [scale_tol_discreterainbow](#), [scale_tol_iridescent](#), [scale_tol_smoothrainbow](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *turku* Sequential Colour Scheme

Usage

```
scale_colour_turku(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_color_turku(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "colour"  
)  
  
scale_fill_turku(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "fill"  
)  
  
scale_edge_colour_turku(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"  
)  
  
scale_edge_color_turku(  
  ...,  
  reverse = FALSE,  
  range = c(0, 1),  
  discrete = FALSE,  
  aesthetics = "edge_colour"
```

```

)

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_YlOrBr`, `scale_tol_discreterainbow`, `scale_tol_iridescent`, `scale_tol_smoothrainbow`

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *vanimo* Diverging Colour Scheme

Usage

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

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

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

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

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

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

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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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 Colour Scheme*

Description

Fabio Crameri's *vik* Diverging Colour Scheme

Usage

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

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

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

```

    discrete = FALSE,
    aesthetics = "fill"
  )

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

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

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

```

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour 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_sunset](#)

Other Fabio Crameri's colour 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_vik0](#)

Examples

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

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

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

scale_crameri_vikO *Fabio Crameri's vikO Cyclic Colour Scheme*

Description

Fabio Crameri's *vikO* Cyclic Colour Scheme

Usage

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

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

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

Arguments

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

Value

A **continuous** scale.

Interpolation

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

Note that the default colour 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 colour schemes: [scale_crameri_bam0](#), [scale_crameri_broc0](#), [scale_crameri_cork0](#), [scale_crameri_roma0](#)

Other Fabio Crameri's colour 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 = 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_logical_discrete`*Binary Discrete Colour Schemes for **ggplot2** and **ggraph***

Description

Provides a colour scale for `logical` values.

Usage

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

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

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

Arguments

<code>...</code>	Arguments passed to <code>ggplot2::discrete_scale()</code> .
<code>aesthetics</code>	A <code>character</code> 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

Examples

```
library(ggplot2)

logic <- data.frame(
  x = rep(1:5, times = 5),
  y = rep(1:5, each = 5),
  values = sample(c(TRUE, FALSE), 25, TRUE)
)

ggplot2::ggplot(logic, ggplot2::aes(x, y, fill = values)) +
  ggplot2::geom_raster() +
  scale_fill_logical()
```

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, aesthetics = "colour")
scale_color_okabeito(..., reverse = FALSE, aesthetics = "colour")
scale_fill_okabeito(..., reverse = FALSE, aesthetics = "fill")
scale_edge_colour_okabeito(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_color_okabeito(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_fill_okabeito(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

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

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 colour-blind safe colour schemes: [scale_crameri_cyclic](#), [scale_crameri_diverging](#), [scale_crameri_mutlisequential](#), [scale_crameri_sequential](#), [scale_tol_discrete](#), [scale_tol_diverging](#), [scale_tol_sequential](#)

Other qualitative colour schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_tol_bright](#), [scale_tol_dark](#), [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()
```

scale_picker

Colour Scale Builder

Description

Builds a colour 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 colour scale function.

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

Value

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

Author(s)

N. Frerebeau

See Also

Other colour 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 Colour Scheme*

Description

Paul Tol's *bright* Discrete Colour Scheme

Usage

```
scale_colour_bright(..., reverse = FALSE, aesthetics = "colour")
scale_color_bright(..., reverse = FALSE, aesthetics = "colour")
scale_fill_bright(..., reverse = FALSE, aesthetics = "fill")
scale_edge_colour_bright(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_color_bright(..., reverse = FALSE, aesthetics = "edge_colour")
scale_edge_fill_bright(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

... Arguments passed to [ggplot2::discrete_scale\(\)](#).

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

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

Value

A [discrete](#) scale.

Interpolation

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

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: [scale_colour_land\(\)](#), [scale_colour_soil\(\)](#), [scale_colour_stratigraphy\(\)](#), [scale_okabeito_discrete](#), [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 colour schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_Yl0rBr](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [scale_tol_highcontrast](#), [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

```
library(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()

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

scale_tol_BuRd *Paul Tol's BuRd Diverging Colour Scheme*

Description

Paul Tol's *BuRd* Diverging Colour 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 logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the colour 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 colour.
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 colour coordinates are linearly interpolated to provide a continuous version of the scheme.

Note that the default colour 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 colour 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_sunset](#)

Other Paul Tol's colour schemes: [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [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_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 Colour Scheme

Description

Paul Tol's *dark* Discrete Colour 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 <code>logical</code> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <code>character</code> 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 colour schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `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

```
library(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()

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

scale_tol_discreterainbow

Paul Tol's discrete rainbow Sequential Colour Scheme

Description

Paul Tol's *discrete rainbow* Sequential Colour 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 <code>logical</code> scalar. Should the resulting vector of colors be reversed?
aesthetics	A <code>character</code> 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 colour 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 colour for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

Rainbow colour 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.
- Colour-blind people have difficulty distinguishing some colours of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two colour schemes that are reasonably clear in colour-blind vision. To remain colour-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 colour 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 colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_Yl0rBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `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(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_highcontrast

Paul Tol's high contrast Discrete Colour Scheme

Description

Paul Tol's *high contrast* Discrete Colour 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 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 colour schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's colour schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [scale_tol_bright](#), [scale_tol_dark](#), [scale_tol_discreterainbow](#), [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

```
library(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()

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

scale_tol_iridescent *Paul Tol's iridescent Sequential Colour Scheme*

Description

Paul Tol's *iridescent* Sequential Colour Scheme

Usage

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

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

scale_fill_iridescent(
  ...,
  reverse = FALSE,
```

```

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

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

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

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

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.
discrete	A logical scalar: should the colour 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 colour.
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 colour 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 colour for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

Rainbow colour 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.
- Colour-blind people have difficulty distinguishing some colours of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two colour schemes that are reasonably clear in colour-blind vision. To remain colour-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 colour 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_Y10rBr`, `scale_tol_discreterainbow`, `scale_tol_smoothrainbow`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_Y10rBr`, `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_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_Y10rBr()
```



```

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 Colour Scheme*

Description

Paul Tol's *light* Discrete Colour 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 <code>ggplot2::discrete_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
aesthetics	A character string or vector of character strings listing the name(s) of the aesthetic(s) that this scale works with.

Value

A [discrete](#) scale.

Interpolation

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

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_iridescent`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
library(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()

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

 scale_tol_mediumcontrast

Paul Tol's medium contrast Discrete Colour Scheme

Description

Paul Tol's *medium contrast* Discrete Colour Scheme

Usage

```
scale_colour_mediumcontrast(..., reverse = FALSE, aesthetics = "colour")
scale_color_mediumcontrast(..., reverse = FALSE, aesthetics = "colour")
scale_fill_mediumcontrast(..., reverse = FALSE, aesthetics = "fill")
scale_edge_colour_mediumcontrast(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
)
scale_edge_color_mediumcontrast(
  ...,
  reverse = FALSE,
  aesthetics = "edge_colour"
)
scale_edge_fill_mediumcontrast(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

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

Value

A **discrete** scale.

Interpolation

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

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_discreterainbow`, `scale_tol_highcontrast`, `scale_tol_iridescent`, `scale_tol_light`, `scale_tol_muted`, `scale_tol_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
library(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()

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

scale_tol_muted *Paul Tol's muted Discrete Colour Scheme*

Description

Paul Tol's *muted* Discrete Colour 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 `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 colour schemes are used as given (no interpolation): colors are picked up to the maximum number of supported values.

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_pale`, `scale_tol_vibrant`

Other Paul Tol's colour 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_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
library(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()

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

scale_tol_pale

Paul Tol's pale Discrete Colour Scheme

Description

Paul Tol's *pale* Discrete Colour Scheme

Usage

```
scale_colour_pale(..., reverse = FALSE, aesthetics = "colour")

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

scale_fill_pale(..., reverse = FALSE, aesthetics = "fill")
```

```
scale_edge_colour_pale(..., reverse = FALSE, aesthetics = "edge_colour")
```

```
scale_edge_color_pale(..., reverse = FALSE, aesthetics = "edge_colour")
```

```
scale_edge_fill_pale(..., reverse = FALSE, aesthetics = "edge_fill")
```

Arguments

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

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

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

Value

A **discrete** scale.

Interpolation

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

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_vibrant`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `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_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

```
library(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()

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

scale_tol_PRGn

Paul Tol's PRGn Diverging Colour Scheme

Description

Paul Tol's *PRGn* Diverging Colour 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 logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.

midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the colour 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 colour.
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 colour coordinates are linearly interpolated to provide a continuous version of the scheme.

Note that the default colour 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 colour 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_sunset`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `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_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`, `scale_tol_vibrant`

Examples

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

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

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

```
ggplot2::ggplot(economics, ggplot2::aes(psavert, pce, colour = unemploy)) +  
  ggplot2::geom_point() +  
  scale_color_PRGn(midpoint = 9000, range = c(0.25, 1))
```

scale_tol_smoothrainbow

Paul Tol's smooth rainbow Sequential Colour Scheme

Description

Paul Tol's *smooth rainbow* Sequential Colour 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_color"  
)
```

```

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

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

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.
discrete	A logical scalar: should the colour 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 colour.
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 colour 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 colour for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

Rainbow colour 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.
- Colour-blind people have difficulty distinguishing some colours of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two colour schemes that are reasonably clear in colour-blind vision. To remain colour-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 colour 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_discreterainbow](#), [scale_tol_iridescent](#)

Other Paul Tol's colour 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_pale](#), [scale_tol_sunset](#), [scale_tol_vibrant](#)

Examples

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

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_Yl0rBr()

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

ggplot2::ggplot(faithfuld, ggplot2::aes(waiting, eruptions, fill = density)) +
  ggplot2::geom_raster() +
  scale_fill_smoothrainbow(range = c(0.25, 1))
```

scale_tol_sunset *Paul Tol's sunset Diverging Colour Scheme*

Description

Paul Tol's *sunset* Diverging Colour 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 logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.
midpoint	A length-one numeric vector giving the midpoint (in data value) of the diverging scale. Defaults to 0.
discrete	A logical scalar: should the colour 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 colour.
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 colour coordinates are linearly interpolated to provide a continuous version of the scheme.

Note that the default colour 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 colour 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](#)

Other Paul Tol's colour schemes: [scale_tol_BuRd](#), [scale_tol_PRGn](#), [scale_tol_YlOrBr](#), [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_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 Colour Scheme*

Description

Paul Tol's *vibrant* Discrete Colour Scheme

Usage

```
scale_colour_vibrant(..., reverse = FALSE, aesthetics = "colour")

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

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

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

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

scale_edge_fill_vibrant(..., reverse = FALSE, aesthetics = "edge_fill")
```


Arguments

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

Value

A **discrete** scale.

Interpolation

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

Qualitative colour schemes

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

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

The pale and dark schemes are not very distinct in either normal or colourblind 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 colour schemes: `scale_colour_land()`, `scale_colour_soil()`, `scale_colour_stratigraphy()`, `scale_okabeito_discrete`, `scale_tol_bright`, `scale_tol_dark`, `scale_tol_highcontrast`, `scale_tol_light`, `scale_tol_mediumcontrast`, `scale_tol_muted`, `scale_tol_pale`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `scale_tol_YlOrBr`, `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_pale`, `scale_tol_smoothrainbow`, `scale_tol_sunset`

Examples

```
library(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()

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

scale_tol_YlOrBr

Paul Tol's YlOrBr Sequential Colour Scheme

Description

Paul Tol's *YlOrBr* Sequential Colour Scheme

Usage

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

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

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

```

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

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

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

```

Arguments

...	Arguments passed to <code>ggplot2::continuous_scale()</code> .
reverse	A logical scalar. Should the resulting vector of colors be reversed?
range	A length-two numeric vector specifying the fraction of the scheme's colour domain to keep.
discrete	A logical scalar: should the colour 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 colour.
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 colour 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 colour for NA can be overridden by passing a value to `ggplot2::continuous_scale()`.

Rainbow colour 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.
- Colour-blind people have difficulty distinguishing some colours of the rainbow.

If such use cannot be avoided, Paul Tol's technical note provides two colour schemes that are reasonably clear in colour-blind vision. To remain colour-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 colour 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_discreterainbow`, `scale_tol_iridescent`, `scale_tol_smoothrainbow`

Other Paul Tol's colour schemes: `scale_tol_BuRd`, `scale_tol_PRGn`, `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_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 colour schemes

- scale_crameri_acton, 17
- scale_crameri_bam, 20
- scale_crameri_bamako, 22
- scale_crameri_bam0, 25
- scale_crameri_batlow, 27
- scale_crameri_batlowK, 30
- scale_crameri_batlowW, 32
- scale_crameri_berlin, 35
- scale_crameri_bilbao, 38
- scale_crameri_broc, 40
- scale_crameri_broc0, 43
- scale_crameri_buda, 45
- scale_crameri_bukavu, 47
- scale_crameri_cork, 49
- scale_crameri_cork0, 52
- scale_crameri_davos, 54
- scale_crameri_devon, 57
- scale_crameri_fes, 59
- scale_crameri_grayC, 61
- scale_crameri_hawaii, 64
- scale_crameri_imola, 67
- scale_crameri_lajolla, 69
- scale_crameri_lapaz, 72
- scale_crameri_lisbon, 75
- scale_crameri_nuuk, 77
- scale_crameri_oleron, 80
- scale_crameri_oslo, 82
- scale_crameri_roma, 85
- scale_crameri_roma0, 87
- scale_crameri_tofino, 89
- scale_crameri_tokyo, 92
- scale_crameri_turku, 95
- scale_crameri_vanimo, 97
- scale_crameri_vik, 100
- scale_crameri_vik0, 103

* Okabe and Ito's colour scheme

- scale_okabeito_discrete, 106

* Paul Tol's colour schemes

- scale_tol_bright, 108
- scale_tol_BuRd, 110
- scale_tol_dark, 112
- scale_tol_discreterainbow, 114
- scale_tol_highcontrast, 116
- scale_tol_iridescent, 118
- scale_tol_light, 121
- scale_tol_mediumcontrast, 123
- scale_tol_muted, 125
- scale_tol_pale, 126
- scale_tol_PRGn, 128
- scale_tol_smoothrainbow, 131
- scale_tol_sunset, 134
- scale_tol_vibrant, 136
- scale_tol_YlOrBr, 138

* color

- colour, 3
- ramp, 11

* colour palettes

- colour, 3
- info, 9
- ramp, 11
- scale_picker, 107

* colour-blind safe colour schemes

- scale_okabeito_discrete, 106

* cyclic colour schemes

- scale_crameri_bam0, 25
- scale_crameri_broc0, 43
- scale_crameri_cork0, 52
- scale_crameri_roma0, 87
- scale_crameri_vik0, 103

* diagnostic tools

- compare, 6
- convert, 8
- plot, 10

* discrete colour schemes

- scale_logical_discrete, 105

* diverging colour schemes

- scale_crameri_bam, 20

- scale_crameri_berlin, 35
- scale_crameri_broc, 40
- scale_crameri_cork, 49
- scale_crameri_lisbon, 75
- scale_crameri_roma, 85
- scale_crameri_tofino, 89
- scale_crameri_vanimo, 97
- scale_crameri_vik, 100
- scale_tol_BuRd, 110
- scale_tol_PRGn, 128
- scale_tol_sunset, 134
- * **multi sequential colour schemes**
 - scale_crameri_bukavu, 47
 - scale_crameri_fes, 59
 - scale_crameri_oleron, 80
- * **qualitative colour schemes**
 - scale_colour_land, 12
 - scale_colour_soil, 14
 - scale_colour_stratigraphy, 15
 - scale_okabeito_discrete, 106
 - scale_tol_bright, 108
 - scale_tol_dark, 112
 - scale_tol_highcontrast, 116
 - scale_tol_light, 121
 - scale_tol_mediumcontrast, 123
 - scale_tol_muted, 125
 - scale_tol_pale, 126
 - scale_tol_vibrant, 136
- * **sequential colour schemes**
 - scale_crameri_acton, 17
 - scale_crameri_bamako, 22
 - scale_crameri_batlow, 27
 - scale_crameri_batlowK, 30
 - scale_crameri_batlowW, 32
 - scale_crameri_bilbao, 38
 - scale_crameri_buda, 45
 - scale_crameri_davos, 54
 - scale_crameri_devon, 57
 - scale_crameri_grayC, 61
 - scale_crameri_hawaii, 64
 - scale_crameri_imola, 67
 - scale_crameri_lajolla, 69
 - scale_crameri_lapaz, 72
 - scale_crameri_nuuk, 77
 - scale_crameri_oslo, 82
 - scale_crameri_tokyo, 92
 - scale_crameri_turku, 95
 - scale_tol_discreterainbow, 114
 - scale_tol_iridescent, 118
 - scale_tol_smoothrainbow, 131
 - scale_tol_YlOrBr, 138
- * **themed colour schemes**
 - scale_colour_land, 12
 - scale_colour_soil, 14
 - scale_colour_stratigraphy, 15
- character, 3, 4, 7, 8, 10, 11, 13, 14, 16, 18, 21, 24, 26, 28, 31, 34, 36, 39, 42, 44, 46, 48, 51, 53, 55, 58, 60, 63, 65, 68, 71, 73, 76, 79, 81, 83, 86, 88, 91, 93, 96, 99, 101, 103, 105–108, 111, 113, 115, 117, 119, 121, 123, 125, 127, 130, 132, 135, 137, 139
- color (colour), 3
- colorRampPalette, 4
- colour, 3, 9, 12, 108
- colour(), 8, 10, 11
- compare, 6, 8, 11
- continuous, 19, 21, 24, 26, 28, 31, 34, 36, 39, 42, 44, 46, 48, 51, 53, 55, 58, 60, 63, 65, 68, 71, 73, 76, 79, 81, 83, 86, 88, 91, 93, 96, 99, 101, 103, 107, 111, 115, 119, 130, 132, 135, 139
- convert, 7, 8, 11
- data.frame, 9
- discrete, 13, 14, 16, 105–108, 113, 117, 121, 123, 125, 127, 137
- distance matrix, 7
- function, 8
- ggplot2::continuous_scale(), 18, 19, 21, 24, 26, 28, 31, 33, 34, 36, 37, 39, 42, 44, 46, 48, 51, 53, 55, 56, 58, 60, 62, 63, 65, 68, 70, 71, 73, 76, 78, 79, 81, 83, 86, 88, 91, 93, 96, 99, 101–104, 111, 115, 119, 120, 129, 130, 132, 135, 139
- ggplot2::discrete_scale(), 13, 14, 16, 105, 106, 108, 113, 117, 121, 123, 125, 127, 137
- graphics::par(), 10
- hist(), 12
- info, 5, 9, 12, 108
- info(), 107

- integer, [4](#), [7](#), [10](#)
- logical, [3](#), [4](#), [7](#), [10](#), [18](#), [21](#), [24](#), [26](#), [28](#), [31](#), [33](#), [34](#), [36](#), [39](#), [42](#), [44](#), [46](#), [48](#), [51](#), [53](#), [55](#), [58](#), [60](#), [62](#), [63](#), [65](#), [68](#), [70](#), [71](#), [73](#), [76](#), [78](#), [79](#), [81](#), [83](#), [86](#), [88](#), [91](#), [93](#), [96](#), [99](#), [101](#), [103](#), [105](#), [106](#), [108](#), [111](#), [113](#), [115](#), [117](#), [119](#), [121](#), [123](#), [125](#), [127](#), [129](#), [130](#), [132](#), [135](#), [137](#), [139](#)
- numeric, [10](#), [11](#), [18](#), [21](#), [24](#), [26](#), [28](#), [31](#), [33](#), [36](#), [39](#), [42](#), [44](#), [46](#), [48](#), [51](#), [53](#), [55](#), [58](#), [60](#), [62](#), [65](#), [68](#), [70](#), [73](#), [76](#), [78](#), [81](#), [83](#), [86](#), [88](#), [91](#), [93](#), [96](#), [99](#), [101](#), [103](#), [111](#), [119](#), [129](#), [130](#), [132](#), [135](#), [139](#)
- plot, [7](#), [8](#), [10](#)
- plot_map (plot), [10](#)
- plot_scheme (plot), [10](#)
- plot_scheme_colorblind (plot), [10](#)
- plot_scheme_colourblind (plot), [10](#)
- plot_tiles (plot), [10](#)
- ramp, [5](#), [9](#), [11](#), [108](#)
- scale_color_acton
(scale_crameri_acton), [17](#)
- scale_color_bam (scale_crameri_bam), [20](#)
- scale_color_bamako
(scale_crameri_bamako), [22](#)
- scale_color_bam0 (scale_crameri_bam0),
[25](#)
- scale_color_batlow
(scale_crameri_batlow), [27](#)
- scale_color_batlowK
(scale_crameri_batlowK), [30](#)
- scale_color_batlowW
(scale_crameri_batlowW), [32](#)
- scale_color_berlin
(scale_crameri_berlin), [35](#)
- scale_color_bilbao
(scale_crameri_bilbao), [38](#)
- scale_color_bright (scale_tol_bright),
[108](#)
- scale_color_broc (scale_crameri_broc),
[40](#)
- scale_color_broc0
(scale_crameri_broc0), [43](#)
- scale_color_buda (scale_crameri_buda),
[45](#)
- scale_color_bukavu
(scale_crameri_bukavu), [47](#)
- scale_color_BuRd (scale_tol_BuRd), [110](#)
- scale_color_cork (scale_crameri_cork),
[49](#)
- scale_color_cork0
(scale_crameri_cork0), [52](#)
- scale_color_dark (scale_tol_dark), [112](#)
- scale_color_davos
(scale_crameri_davos), [54](#)
- scale_color_devon
(scale_crameri_devon), [57](#)
- scale_color_discreterainbow
(scale_tol_discreterainbow),
[114](#)
- scale_color_fes (scale_crameri_fes), [59](#)
- scale_color_grayC
(scale_crameri_grayC), [61](#)
- scale_color_hawaii
(scale_crameri_hawaii), [64](#)
- scale_color_highcontrast
(scale_tol_highcontrast), [116](#)
- scale_color_imola
(scale_crameri_imola), [67](#)
- scale_color_iridescent
(scale_tol_iridescent), [118](#)
- scale_color_lajolla
(scale_crameri_lajolla), [69](#)
- scale_color_land (scale_colour_land), [12](#)
- scale_color_lapaz
(scale_crameri_lapaz), [72](#)
- scale_color_light (scale_tol_light), [121](#)
- scale_color_lisbon
(scale_crameri_lisbon), [75](#)
- scale_color_logical
(scale_logical_discrete), [105](#)
- scale_color_mediumcontrast
(scale_tol_mediumcontrast), [123](#)
- scale_color_muted (scale_tol_muted), [125](#)
- scale_color_nuuk (scale_crameri_nuuk),
[77](#)
- scale_color_okabeito
(scale_okabeito_discrete), [106](#)
- scale_color_oleron
(scale_crameri_oleron), [80](#)
- scale_color_oslo (scale_crameri_oslo),
[82](#)
- scale_color_pale (scale_tol_pale), [126](#)

- scale_color_picker (scale_picker), 107
- scale_color_PRGN (scale_tol_PRGN), 128
- scale_color_roma (scale_crameri_roma), 85
- scale_color_roma0 (scale_crameri_roma0), 87
- scale_color_smoothrainbow (scale_tol_smoothrainbow), 131
- scale_color_soil (scale_colour_soil), 14
- scale_color_stratigraphy (scale_colour_stratigraphy), 15
- scale_color_sunset (scale_tol_sunset), 134
- scale_color_tofino (scale_crameri_tofino), 89
- scale_color_tokyo (scale_crameri_tokyo), 92
- scale_color_turku (scale_crameri_turku), 95
- scale_color_vanimo (scale_crameri_vanimo), 97
- scale_color_vibrant (scale_tol_vibrant), 136
- scale_color_vik (scale_crameri_vik), 100
- scale_color_vik0 (scale_crameri_vik0), 103
- scale_color_YlOrBr (scale_tol_YlOrBr), 138
- scale_colour_acton (scale_crameri_acton), 17
- scale_colour_bam (scale_crameri_bam), 20
- scale_colour_bamako (scale_crameri_bamako), 22
- scale_colour_bam0 (scale_crameri_bam0), 25
- scale_colour_batlow (scale_crameri_batlow), 27
- scale_colour_batlowK (scale_crameri_batlowK), 30
- scale_colour_batlowW (scale_crameri_batlowW), 32
- scale_colour_berlin (scale_crameri_berlin), 35
- scale_colour_bilbao (scale_crameri_bilbao), 38
- scale_colour_bright (scale_tol_bright), 108
- scale_colour_broc (scale_crameri_broc), 40
- scale_colour_broc0 (scale_crameri_broc0), 43
- scale_colour_buda (scale_crameri_buda), 45
- scale_colour_bukavu (scale_crameri_bukavu), 47
- scale_colour_BuRd (scale_tol_BuRd), 110
- scale_colour_cork (scale_crameri_cork), 49
- scale_colour_cork0 (scale_crameri_cork0), 52
- scale_colour_dark (scale_tol_dark), 112
- scale_colour_davos (scale_crameri_davos), 54
- scale_colour_devon (scale_crameri_devon), 57
- scale_colour_discreterainbow (scale_tol_discreterainbow), 114
- scale_colour_fes (scale_crameri_fes), 59
- scale_colour_grayC (scale_crameri_grayC), 61
- scale_colour_hawaii (scale_crameri_hawaii), 64
- scale_colour_highcontrast (scale_tol_highcontrast), 116
- scale_colour_imola (scale_crameri_imola), 67
- scale_colour_iridescent (scale_tol_iridescent), 118
- scale_colour_lajolla (scale_crameri_lajolla), 69
- scale_colour_land, 12, 15, 16, 107, 109, 113, 117, 122, 124, 126, 128, 137
- scale_colour_lapaz (scale_crameri_lapaz), 72
- scale_colour_light (scale_tol_light), 121
- scale_colour_lisbon (scale_crameri_lisbon), 75
- scale_colour_logical (scale_logical_discrete), 105
- scale_colour_mediumcontrast (scale_tol_mediumcontrast), 123
- scale_colour_muted (scale_tol_muted), 125
- scale_colour_nuuk (scale_crameri_nuuk),

- 77
- scale_colour_okabeito
(scale_okabeito_discrete), 106
- scale_colour_oleron
(scale_crameri_oleron), 80
- scale_colour_oslo (scale_crameri_oslo),
82
- scale_colour_pale (scale_tol_pale), 126
- scale_colour_picker (scale_picker), 107
- scale_colour_PRGn (scale_tol_PRGn), 128
- scale_colour_roma (scale_crameri_roma),
85
- scale_colour_roma0
(scale_crameri_roma0), 87
- scale_colour_smoothrainbow
(scale_tol_smoothrainbow), 131
- scale_colour_soil, 13, 14, 16, 107, 109,
113, 117, 122, 124, 126, 128, 137
- scale_colour_stratigraphy, 13, 15, 15,
107, 109, 113, 117, 122, 124, 126,
128, 137
- scale_colour_sunset (scale_tol_sunset),
134
- scale_colour_tofino
(scale_crameri_tofino), 89
- scale_colour_tokyo
(scale_crameri_tokyo), 92
- scale_colour_turku
(scale_crameri_turku), 95
- scale_colour_vanimo
(scale_crameri_vanimo), 97
- scale_colour_vibrant
(scale_tol_vibrant), 136
- scale_colour_vik (scale_crameri_vik),
100
- scale_colour_vik0 (scale_crameri_vik0),
103
- scale_colour_Yl0rBr (scale_tol_Yl0rBr),
138
- scale_crameri_acton, 17, 22, 24, 26, 29, 32,
34, 37, 40, 42, 44, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 94, 97, 99, 102, 104,
116, 120, 133, 140
- scale_crameri_bam, 19, 20, 24, 26, 29, 32,
34, 37, 40, 42, 44, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 94, 97, 99, 102, 104,
112, 130, 136
- scale_crameri_bamako, 19, 22, 22, 26, 29,
32, 34, 37, 40, 42, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 94, 97, 99, 102,
104, 116, 120, 133, 140
- scale_crameri_bam0, 19, 22, 24, 25, 29, 32,
34, 37, 40, 42, 44, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 94, 97, 99, 102, 104
- scale_crameri_batlow, 19, 22, 24, 26, 27,
32, 34, 37, 40, 42, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 94, 97, 99, 102,
104, 116, 120, 133, 140
- scale_crameri_batlowK, 19, 22, 24, 26, 29,
30, 34, 37, 40, 42, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 94, 97, 99, 102,
104, 116, 120, 133, 140
- scale_crameri_batlowW, 19, 22, 24, 26, 29,
32, 32, 37, 40, 42, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 94, 97, 99, 102,
104, 116, 120, 133, 140
- scale_crameri_berlin, 19, 22, 24, 26, 29,
32, 34, 35, 40, 42, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 94, 97, 99, 102,
104, 112, 130, 136
- scale_crameri_bilbao, 19, 22, 24, 26, 29,
32, 34, 37, 38, 42, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 94, 97, 99, 102,
104, 116, 120, 133, 140
- scale_crameri_broc, 19, 22, 24, 26, 29, 32,
34, 37, 40, 40, 44, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 94, 97, 99, 102, 104,
112, 130, 136
- scale_crameri_broc0, 19, 22, 24, 26, 29, 32,
34, 37, 40, 42, 43, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 94, 97, 99, 102, 104
- scale_crameri_buda, 19, 22, 24, 26, 29, 32,
34, 37, 40, 42, 44, 45, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 94, 97, 99, 102, 104,

- 116, 120, 133, 140
- scale_crameri_bukavu, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104
- scale_crameri_cork, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 49, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 112, 130, 136
- scale_crameri_cork0, 19, 22, 24, 26, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 52, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104
- scale_crameri_cyclic, 107
- scale_crameri_davos, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 54, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_devon, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 57, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_diverging, 107
- scale_crameri_fes, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 59, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104
- scale_crameri_grayC, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 61, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_hawaii, 19, 22, 24, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 63, 64, 69, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_imola, 19, 22, 24, 25, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 67, 71, 74, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_lajolla, 19, 22, 24, 25, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 69, 74, 77, 79, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_lapaz, 19, 22, 24, 25, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 72, 77, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 116, 120, 133, 140
- scale_crameri_lisbon, 19, 22, 25, 27, 29, 32, 34, 37, 40, 42, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 75, 79, 81, 84, 87, 89, 91, 94, 97, 99, 102, 104, 112, 130, 136
- scale_crameri_mutlisequential, 107
- scale_crameri_nuuk, 19, 22, 24, 25, 27, 29, 32, 34, 37, 40, 43, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 77, 81, 84, 87, 89, 92, 94, 97, 100, 102, 104, 116, 120, 133, 140
- scale_crameri_oleron, 19, 22, 25, 27, 29, 32, 34, 37, 40, 43, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 80, 84, 87, 89, 92, 94, 97, 100, 102, 104
- scale_crameri_oslo, 19, 22, 24, 25, 27, 29, 32, 34, 37, 40, 43, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 82, 87, 89, 92, 94, 97, 100, 102, 104, 116, 120, 133, 140
- scale_crameri_roma, 19, 22, 25, 27, 29, 32, 34, 37, 40, 42–44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 85, 89, 91, 92, 94, 97, 99, 100, 102, 104, 112, 130, 136
- scale_crameri_roma0, 19, 22, 25–27, 29, 32, 34, 37, 40, 43, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 87, 92, 94, 97, 100, 102, 104
- scale_crameri_sequential, 107
- scale_crameri_tofino, 19, 22, 25, 27, 29, 32, 34, 37, 40, 42–44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79, 81, 84, 87, 89, 89, 94, 97, 99, 100, 102, 104, 112, 130, 136
- scale_crameri_tokyo, 19, 22, 24, 25, 27, 29, 32, 34, 37, 40, 43, 44, 47, 49, 52, 54, 56, 59, 61, 63, 66, 69, 71, 74, 77, 79,

- 81, 84, 87, 89, 92, 92, 97, 100, 102,
104, 116, 120, 133, 140
- scale_crameri_turku, 19, 22, 24, 25, 27, 29,
32, 34, 37, 40, 43, 44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 92, 94, 95, 100, 102,
104, 116, 120, 133, 140
- scale_crameri_vanimo, 19, 22, 25, 27, 29,
32, 34, 37, 40, 42–44, 47, 49, 52, 54,
56, 59, 61, 63, 66, 69, 71, 74, 77, 79,
81, 84, 87, 89, 91, 92, 94, 97, 97,
102, 104, 112, 130, 136
- scale_crameri_vik, 19, 22, 25, 27, 29, 32,
34, 37, 40, 42–44, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 91, 92, 94, 97, 99, 100,
100, 104, 112, 130, 136
- scale_crameri_vik0, 19, 22, 25–27, 29, 32,
34, 37, 40, 43, 44, 47, 49, 52, 54, 56,
59, 61, 63, 66, 69, 71, 74, 77, 79, 81,
84, 87, 89, 92, 94, 97, 100, 102, 103
- scale_edge_color_acton
(scale_crameri_acton), 17
- scale_edge_color_bam
(scale_crameri_bam), 20
- scale_edge_color_bamako
(scale_crameri_bamako), 22
- scale_edge_color_batlow
(scale_crameri_batlow), 27
- scale_edge_color_batlowK
(scale_crameri_batlowK), 30
- scale_edge_color_batlowW
(scale_crameri_batlowW), 32
- scale_edge_color_berlin
(scale_crameri_berlin), 35
- scale_edge_color_bilbao
(scale_crameri_bilbao), 38
- scale_edge_color_bright
(scale_tol_bright), 108
- scale_edge_color_broc
(scale_crameri_broc), 40
- scale_edge_color_buda
(scale_crameri_buda), 45
- scale_edge_color_BuRd (scale_tol_BuRd),
110
- scale_edge_color_cork
(scale_crameri_cork), 49
- scale_edge_color_dark (scale_tol_dark),
112
- scale_edge_color_davos
(scale_crameri_davos), 54
- scale_edge_color_devon
(scale_crameri_devon), 57
- scale_edge_color_discreterainbow
(scale_tol_discreterainbow),
114
- scale_edge_color_grayC
(scale_crameri_grayC), 61
- scale_edge_color_hawaii
(scale_crameri_hawaii), 64
- scale_edge_color_highcontrast
(scale_tol_highcontrast), 116
- scale_edge_color_imola
(scale_crameri_imola), 67
- scale_edge_color_iridescent
(scale_tol_iridescent), 118
- scale_edge_color_lajolla
(scale_crameri_lajolla), 69
- scale_edge_color_land
(scale_colour_land), 12
- scale_edge_color_lapaz
(scale_crameri_lapaz), 72
- scale_edge_color_light
(scale_tol_light), 121
- scale_edge_color_lisbon
(scale_crameri_lisbon), 75
- scale_edge_color_mediumcontrast
(scale_tol_mediumcontrast), 123
- scale_edge_color_muted
(scale_tol_muted), 125
- scale_edge_color_nuuk
(scale_crameri_nuuk), 77
- scale_edge_color_okabeito
(scale_okabeito_discrete), 106
- scale_edge_color_oslo
(scale_crameri_oslo), 82
- scale_edge_color_pale (scale_tol_pale),
126
- scale_edge_color_picker (scale_picker),
107
- scale_edge_color_PRGn (scale_tol_PRGn),
128
- scale_edge_color_roma
(scale_crameri_roma), 85
- scale_edge_color_smoothrainbow
(scale_tol_smoothrainbow), 131

- scale_edge_color_soil
 - (scale_colour_soil), 14
- scale_edge_color_stratigraphy
 - (scale_colour_stratigraphy), 15
- scale_edge_color_sunset
 - (scale_tol_sunset), 134
- scale_edge_color_tofino
 - (scale_crameri_tofino), 89
- scale_edge_color_tokyo
 - (scale_crameri_tokyo), 92
- scale_edge_color_turku
 - (scale_crameri_turku), 95
- scale_edge_color_vanimo
 - (scale_crameri_vanimo), 97
- scale_edge_color_vibrant
 - (scale_tol_vibrant), 136
- scale_edge_color_vik
 - (scale_crameri_vik), 100
- scale_edge_color_YlOrBr
 - (scale_tol_YlOrBr), 138
- scale_edge_colour_acton
 - (scale_crameri_acton), 17
- scale_edge_colour_bam
 - (scale_crameri_bam), 20
- scale_edge_colour_bamako
 - (scale_crameri_bamako), 22
- scale_edge_colour_batlow
 - (scale_crameri_batlow), 27
- scale_edge_colour_batlowK
 - (scale_crameri_batlowK), 30
- scale_edge_colour_batlowW
 - (scale_crameri_batlowW), 32
- scale_edge_colour_berlin
 - (scale_crameri_berlin), 35
- scale_edge_colour_bilbao
 - (scale_crameri_bilbao), 38
- scale_edge_colour_bright
 - (scale_tol_bright), 108
- scale_edge_colour_broc
 - (scale_crameri_broc), 40
- scale_edge_colour_buda
 - (scale_crameri_buda), 45
- scale_edge_colour_BuRd
 - (scale_tol_BuRd), 110
- scale_edge_colour_cork
 - (scale_crameri_cork), 49
- scale_edge_colour_dark
 - (scale_tol_dark), 112
- scale_edge_colour_davos
 - (scale_crameri_davos), 54
- scale_edge_colour_devon
 - (scale_crameri_devon), 57
- scale_edge_colour_discreterainbow
 - (scale_tol_discreterainbow), 114
- scale_edge_colour_grayC
 - (scale_crameri_grayC), 61
- scale_edge_colour_hawaii
 - (scale_crameri_hawaii), 64
- scale_edge_colour_highcontrast
 - (scale_tol_highcontrast), 116
- scale_edge_colour_imola
 - (scale_crameri_imola), 67
- scale_edge_colour_iridescent
 - (scale_tol_iridescent), 118
- scale_edge_colour_lajolla
 - (scale_crameri_lajolla), 69
- scale_edge_colour_land
 - (scale_colour_land), 12
- scale_edge_colour_lapaz
 - (scale_crameri_lapaz), 72
- scale_edge_colour_light
 - (scale_tol_light), 121
- scale_edge_colour_lisbon
 - (scale_crameri_lisbon), 75
- scale_edge_colour_mediumcontrast
 - (scale_tol_mediumcontrast), 123
- scale_edge_colour_muted
 - (scale_tol_muted), 125
- scale_edge_colour_nuuk
 - (scale_crameri_nuuk), 77
- scale_edge_colour_okabeito
 - (scale_okabeito_discrete), 106
- scale_edge_colour_oslo
 - (scale_crameri_oslo), 82
- scale_edge_colour_pale
 - (scale_tol_pale), 126
- scale_edge_colour_picker
 - (scale_picker), 107
- scale_edge_colour_PRGn
 - (scale_tol_PRGn), 128
- scale_edge_colour_roma
 - (scale_crameri_roma), 85
- scale_edge_colour_smoothrainbow
 - (scale_tol_smoothrainbow), 131
- scale_edge_colour_soil

- (scale_colour_soil), 14
- scale_edge_colour_stratigraphy
(scale_colour_stratigraphy), 15
- scale_edge_colour_sunset
(scale_tol_sunset), 134
- scale_edge_colour_tofino
(scale_crameri_tofino), 89
- scale_edge_colour_tokyo
(scale_crameri_tokyo), 92
- scale_edge_colour_turku
(scale_crameri_turku), 95
- scale_edge_colour_vanimo
(scale_crameri_vanimo), 97
- scale_edge_colour_vibrant
(scale_tol_vibrant), 136
- scale_edge_colour_vik
(scale_crameri_vik), 100
- scale_edge_colour_YlOrBr
(scale_tol_YlOrBr), 138
- scale_edge_fill_acton
(scale_crameri_acton), 17
- scale_edge_fill_bam
(scale_crameri_bam), 20
- scale_edge_fill_bamako
(scale_crameri_bamako), 22
- scale_edge_fill_batlow
(scale_crameri_batlow), 27
- scale_edge_fill_batlowK
(scale_crameri_batlowK), 30
- scale_edge_fill_batlowW
(scale_crameri_batlowW), 32
- scale_edge_fill_berlin
(scale_crameri_berlin), 35
- scale_edge_fill_bilbao
(scale_crameri_bilbao), 38
- scale_edge_fill_bright
(scale_tol_bright), 108
- scale_edge_fill_broc
(scale_crameri_broc), 40
- scale_edge_fill_buda
(scale_crameri_buda), 45
- scale_edge_fill_BuRd (scale_tol_BuRd),
110
- scale_edge_fill_cork
(scale_crameri_cork), 49
- scale_edge_fill_dark (scale_tol_dark),
112
- scale_edge_fill_davos
(scale_crameri_davos), 54
- scale_edge_fill_devon
(scale_crameri_devon), 57
- scale_edge_fill_discreterainbow
(scale_tol_discreterainbow),
114
- scale_edge_fill_grayC
(scale_crameri_grayC), 61
- scale_edge_fill_hawaii
(scale_crameri_hawaii), 64
- scale_edge_fill_highcontrast
(scale_tol_highcontrast), 116
- scale_edge_fill_imola
(scale_crameri_imola), 67
- scale_edge_fill_iridescent
(scale_tol_iridescent), 118
- scale_edge_fill_lajolla
(scale_crameri_lajolla), 69
- scale_edge_fill_land
(scale_colour_land), 12
- scale_edge_fill_lapaz
(scale_crameri_lapaz), 72
- scale_edge_fill_light
(scale_tol_light), 121
- scale_edge_fill_lisbon
(scale_crameri_lisbon), 75
- scale_edge_fill_mediumcontrast
(scale_tol_mediumcontrast), 123
- scale_edge_fill_muted
(scale_tol_muted), 125
- scale_edge_fill_nuuk
(scale_crameri_nuuk), 77
- scale_edge_fill_okabeito
(scale_okabeito_discrete), 106
- scale_edge_fill_oslo
(scale_crameri_oslo), 82
- scale_edge_fill_pale (scale_tol_pale),
126
- scale_edge_fill_picker (scale_picker),
107
- scale_edge_fill_PRGn (scale_tol_PRGn),
128
- scale_edge_fill_roma
(scale_crameri_roma), 85
- scale_edge_fill_smoothrainbow
(scale_tol_smoothrainbow), 131
- scale_edge_fill_soil
(scale_colour_soil), 14

- scale_edge_fill_stratigraphy
(scale_colour_stratigraphy), 15
- scale_edge_fill_sunset
(scale_tol_sunset), 134
- scale_edge_fill_tofino
(scale_crameri_tofino), 89
- scale_edge_fill_tokyo
(scale_crameri_tokyo), 92
- scale_edge_fill_turku
(scale_crameri_turku), 95
- scale_edge_fill_vanimo
(scale_crameri_vanimo), 97
- scale_edge_fill_vibrant
(scale_tol_vibrant), 136
- scale_edge_fill_vik
(scale_crameri_vik), 100
- scale_edge_fill_YlOrBr
(scale_tol_YlOrBr), 138
- scale_fill_acton (scale_crameri_acton),
17
- scale_fill_bam (scale_crameri_bam), 20
- scale_fill_bamako
(scale_crameri_bamako), 22
- scale_fill_bam0 (scale_crameri_bam0), 25
- scale_fill_batlow
(scale_crameri_batlow), 27
- scale_fill_batlowK
(scale_crameri_batlowK), 30
- scale_fill_batlowW
(scale_crameri_batlowW), 32
- scale_fill_berlin
(scale_crameri_berlin), 35
- scale_fill_bilbao
(scale_crameri_bilbao), 38
- scale_fill_bright (scale_tol_bright),
108
- scale_fill_broc (scale_crameri_broc), 40
- scale_fill_broc0 (scale_crameri_broc0),
43
- scale_fill_buda (scale_crameri_buda), 45
- scale_fill_bukavu
(scale_crameri_bukavu), 47
- scale_fill_BuRd (scale_tol_BuRd), 110
- scale_fill_cork (scale_crameri_cork), 49
- scale_fill_cork0 (scale_crameri_cork0),
52
- scale_fill_dark (scale_tol_dark), 112
- scale_fill_davos (scale_crameri_davos),
54
- scale_fill_devon (scale_crameri_devon),
57
- scale_fill_discreterainbow
(scale_tol_discreterainbow),
114
- scale_fill_fes (scale_crameri_fes), 59
- scale_fill_grayC (scale_crameri_grayC),
61
- scale_fill_hawaii
(scale_crameri_hawaii), 64
- scale_fill_highcontrast
(scale_tol_highcontrast), 116
- scale_fill_imola (scale_crameri_imola),
67
- scale_fill_iridescent
(scale_tol_iridescent), 118
- scale_fill_lajolla
(scale_crameri_lajolla), 69
- scale_fill_land (scale_colour_land), 12
- scale_fill_lapaz (scale_crameri_lapaz),
72
- scale_fill_light (scale_tol_light), 121
- scale_fill_lisbon
(scale_crameri_lisbon), 75
- scale_fill_logical
(scale_logical_discrete), 105
- scale_fill_mediumcontrast
(scale_tol_mediumcontrast), 123
- scale_fill_nuuk (scale_crameri_nuuk), 77
- scale_fill_okabeito
(scale_okabeito_discrete), 106
- scale_fill_oleron
(scale_crameri_oleron), 80
- scale_fill_oslo (scale_crameri_oslo), 82
- scale_fill_pale (scale_tol_pale), 126
- scale_fill_picker (scale_picker), 107
- scale_fill_PRGn (scale_tol_PRGn), 128
- scale_fill_roma (scale_crameri_roma), 85
- scale_fill_roma0 (scale_crameri_roma0),
87
- scale_fill_smoothrainbow
(scale_tol_smoothrainbow), 131
- scale_fill_soil (scale_colour_soil), 14
- scale_fill_stratigraphy
(scale_colour_stratigraphy), 15
- scale_fill_sunset (scale_tol_sunset),
134

- scale_fill_tofino
(scale_crameri_tofino), 89
- scale_fill_tokyo (scale_crameri_tokyo),
92
- scale_fill_turku (scale_crameri_turku),
95
- scale_fill_vanimo
(scale_crameri_vanimo), 97
- scale_fill_vibrant (scale_tol_vibrant),
136
- scale_fill_vik (scale_crameri_vik), 100
- scale_fill_vik0 (scale_crameri_vik0),
103
- scale_fill_YlOrBr (scale_tol_YlOrBr),
138
- scale_logical_discrete, 105
- scale_okabeito_discrete, 13, 15, 16, 106,
109, 113, 117, 122, 124, 126, 128,
137
- scale_picker, 5, 9, 12, 107
- scale_tol_bright, 13, 15, 16, 107, 108, 112,
113, 116–118, 120, 122, 124, 126,
128, 130, 133, 136, 137, 140
- scale_tol_BuRd, 22, 37, 42, 52, 77, 87, 91,
99, 102, 109, 110, 113, 116, 118,
120, 122, 124, 126, 128, 130, 133,
136, 137, 140
- scale_tol_dark, 13, 15, 16, 107, 109, 112,
112, 116–118, 120, 122, 124, 126,
128, 130, 133, 136, 137, 140
- scale_tol_discrete, 107
- scale_tol_discreterainbow, 19, 24, 29, 32,
34, 40, 47, 56, 59, 63, 66, 69, 71, 74,
79, 84, 94, 97, 109, 112, 113, 114,
118, 120, 122, 124, 126, 128, 130,
133, 136, 137, 140
- scale_tol_diverging, 107
- scale_tol_highcontrast, 13, 15, 16, 107,
109, 112, 113, 116, 116, 120, 122,
124, 126, 128, 130, 133, 136, 137,
140
- scale_tol_iridescent, 19, 24, 29, 32, 34,
40, 47, 56, 59, 63, 66, 69, 71, 74, 79,
84, 94, 97, 109, 112, 113, 116, 118,
118, 122, 124, 126, 128, 130, 133,
136, 137, 140
- scale_tol_light, 13, 15, 16, 107, 109, 112,
113, 116–118, 120, 121, 124, 126,
128, 130, 133, 136, 137, 140
- scale_tol_mediumcontrast, 13, 15, 16, 107,
109, 112, 113, 116–118, 120, 122,
123, 126, 128, 130, 133, 136, 137,
140
- scale_tol_muted, 13, 15, 16, 107, 109, 112,
113, 116–118, 120, 122, 124, 125,
128, 130, 133, 136, 137, 140
- scale_tol_pale, 13, 15, 16, 107, 109, 112,
113, 116–118, 120, 122, 124, 126,
126, 130, 133, 136, 137, 140
- scale_tol_PRGn, 22, 37, 42, 52, 77, 87, 91,
99, 102, 109, 112, 113, 116, 118,
120, 122, 124, 126, 128, 128, 133,
136, 137, 140
- scale_tol_sequential, 107
- scale_tol_smoothrainbow, 19, 24, 29, 32,
34, 40, 47, 56, 59, 63, 66, 69, 71, 74,
79, 84, 94, 97, 109, 112, 113, 116,
118, 120, 122, 124, 126, 128, 130,
131, 136, 137, 140
- scale_tol_sunset, 22, 37, 42, 52, 77, 87, 91,
99, 102, 109, 112, 113, 116, 118,
120, 122, 124, 126, 128, 130, 133,
134, 137, 140
- scale_tol_vibrant, 13, 15, 16, 107, 109,
112, 113, 116–118, 120, 122, 124,
126, 128, 130, 133, 136, 136, 140
- scale_tol_YlOrBr, 19, 24, 29, 32, 34, 40, 47,
56, 59, 63, 66, 69, 71, 74, 79, 84, 94,
97, 109, 112, 113, 116, 118, 120,
122, 124, 126, 128, 130, 133, 136,
137, 138
- spacesXYZ: :DeltaE(), 7