Package ‘ggthemes’

November 21, 2023

Version  5.0.0
Title    Extra Themes, Scales and Geoms for 'ggplot2'
Depends  R (>= 3.3.0)
Imports  ggplot2 (>= 3.0.0), graphics, grid, lifecycle, methods, purrr,
          scales, stringr, tibble
Suggests dplyr, covr, extrafont, glue, knitr, lattice, lintr, maps,
         mapproj, pander, quantreg, rlang, rmarkdown, spelling, testthat
         (>= 3.2.0), tidyrr, withr, V8
Description Some extra themes, geoms, and scales for 'ggplot2'.
          Provides 'ggplot2' themes and scales that replicate the look of plots
          by Edward Tufte, Stephen Few, 'Fivethirtyeight', 'The Economist', 'Stata',
          'Excel', and 'The Wall Street Journal', among others.
          Provides 'geoms' for Tufte’s box plot and range frame.
License   GPL-2
URL       https://github.com/jrnold/ggthemes
BugReports https://github.com/jrnold/ggthemes
RoxygenNote 7.2.3
LazyData  true
Language en-US
Encoding UTF-8
Config/testthat/edition 3
NeedsCompilation no
Author  Jeffrey B. Arnold [aut, cre] (<https://orcid.org/0000-0001-9953-3904>),
          Gergely Daroczi [ctb],
          Bo Werth [ctb],
          Brian Weitzner [ctb],
          Joshua Kunst [ctb],
          Baptise Auguie [ctb],
          Bob Rudis [ctb],
          Hadley Wickham [ctb] (Code from the ggplot2 package.),
          Justin Talbot [ctb] (Code from the labeling package),
          Joshua London [ctb]
Maintainer  Jeffrey B. Arnold <jeffrey.arnold@gmail.com>
Repository  CRAN
Date/Publication  2023-11-21 10:50:02 UTC

R topics documented:

bank_slopes ........................................... 4
calc_pal .................................................. 6
calc_shape_pal ........................................ 6
canva_pal ............................................... 7
canva_palettes ........................................ 8
circlefill_shape_pal .................................. 9
cleveland_shape_pal .................................... 9
colorblind_pal ......................................... 11
economist_pal ......................................... 12
eexcel_new_pal ......................................... 13
eexcel_pal ................................................ 14
extended_range_breaks ................................. 14
few_pal .................................................... 16
few_shape_pal .......................................... 17
fivethirtyeight_pal ..................................... 17
gdocs_pal ............................................... 18
geom_rangeframe ....................................... 18
geom_tufteboxplot .................................... 20
ggthemes ................................................ 23
ggthemes_data ......................................... 24
hc_pal ...................................................... 24
palette_pander ......................................... 25
ptol_pal ................................................... 25
scale_color_pander ..................................... 26
scale_colour_canva .................................... 27
scale_colour_economist ............................... 28
scale_colour_excel_new ............................... 29
scale_colour_few ....................................... 31
scale_colour_fivethirtyeight ......................... 33
scale_colour_gradient2_tableau ........................ 34
scale_colour_gradient_tableau ......................... 36
scale_colour_hc ......................................... 38
scale_colour_ptol ...................................... 40
scale_colour_stata ..................................... 41
scale_colour_tableau ................................... 43
scale_colour_wsj ....................................... 44
scale_fill_calc ......................................... 46
scale_fill_excel ........................................ 47
scale_fill_gdocs ....................................... 49
scale_fill_solarized ................................... 51
scale_linetype_stata ................................... 52
R topics documented:

- scale_shape_calc ........................................ 54
- scale_shape_circlefill ................................. 55
- scale_shape_cleveland ................................ 57
- scale_shape_few ......................................... 58
- scale_shape_stata ....................................... 59
- scale_shape_tableau .................................... 60
- scale_shape_tremmel .................................. 62
- show_linetypes ......................................... 63
- show_shapes ............................................. 64
- smart_digits ............................................ 65
- solarized_pal ............................................ 66
- stata_linetype_pal ..................................... 66
- stata_pal ................................................ 67
- stata_shape_pal ......................................... 67
- stat_fivenumber ......................................... 68
- tableau_color_pal ....................................... 69
- tableau_gradient_pal ................................... 71
- tableau_shape_pal ...................................... 72
- theme_base ............................................... 73
- theme_calc .............................................. 74
- theme_clean ............................................. 74
- theme_economist ........................................ 75
- theme_excel ............................................. 77
- theme_excel_new ................................-------- 78
- theme_few ............................................... 79
- theme_fivethirtyeight ................................. 80
- theme.foundation ....................................... 80
- theme_gdocs ............................................. 81
- theme_hc ................................................ 81
- theme_igray ............................................. 83
- theme_map ............................................... 84
- theme_pander ........................................... 84
- theme_par ............................................... 86
- theme_solarized ........................................ 87
- theme_solid ............................................. 88
- theme_stata ............................................. 88
- theme_tufte ............................................ 90
- theme_wsj ............................................... 91
- tremmel_shape_pal ..................................... 92
- wsj_pal .................................................. 93

Index ...................................................... 94
Description

Calculate the optimal aspect ratio of a line graph by banking the slopes to 45 degrees as suggested by W.S. Cleveland. This maximizes the ability to visually differentiate differences in slope. This function will calculate the optimal aspect ratio for a line plot using any of the methods described in Herr and Argwala (2006). In their review of the methods they suggest using median absolute slope banking (‘ms’), which produces aspect ratios which are generally the median of the various methods provided here.

Usage

bank_slopes(x, y, cull = FALSE, weight = NULL, method = c("ms", "as"), ...)

Arguments

- x: x values
- y: y values
- cull: logical. Remove all slopes of 0 or Inf.
- weight: No longer used, but kept for backwards compatibility.
- method: One of 'ms' (Median Absolute Slope) or 'as' (Average Absolute Slope). Other options are no longer supported, and will use 'ms' instead with a warning.
- ...: No longer used, but kept for backwards compatibility.

Value

numeric The aspect ratio (x, y).

Methods

As written, all of these methods calculate the aspect ratio (x / y), but bank_slopes will return (y / x) to be compatible with link[ggplot2]{coord_fixed()).

Median Absolute Slopes Banking

Let the aspect ratio be \( \alpha = \frac{w}{h} \) then the median absolute slope banking is the \( \alpha \) such that,

\[
\text{median } \left| \frac{s_i}{\alpha} \right| = 1
\]

Let \( R_z = z_{\text{max}} - z_{\text{min}} \) for \( z = x, y, \) and \( M = \text{median} \| s_i \|. \) Then,

\[
\alpha = M \frac{R_x}{R_y}
\]

Average Absolute Slope Banking
Let the aspect ratio be $\alpha = \frac{w}{h}$. Then the mean absolute slope banking is the $\alpha$ such that,

$$mean\left|\frac{s}{\alpha}\right| = 1$$

Heer and Agrawala (2006) and Cleveland discuss several other methods including average (weighted) orientation, and global and local orientation resolution. These are no longer implemented in this function. In general, either the median or average absolute slopes will produce reasonable results without requiring optimization.

References


See Also

banking()

Examples

```r
library("ggplot2")

# Use the classic sunspot data from Cleveland's original paper
x <- seq_along(sunspot.year)
y <- as.numeric(sunspot.year)
# Without banking
m <- ggplot(data.frame(x = x, y = y), aes(x = x, y = y)) +
  geom_line()

m

## Using the default method, Median Absolute Slope
ratio <- bank_slopes(x, y)
m + coord_fixed(ratio = ratio)

## Using culling
## Average Absolute Slope
bank_slopes(x, y, method = "as")
```

```
calc_pal  
**Calc color palette (discrete)**

**Description**
Color palettes from LibreOffice Calc. This palette has 12 values.

**Usage**
calc_pal()

**See Also**
Other colour calc: `scale_fill_calc()`

**Examples**
```r
library("scales")
show_col(calc_pal()(12))
```

calc_shape_pal  
**Calc shape palette (discrete)**

**Description**
Shape palette based on the shapes used in LibreOffice Calc.

**Usage**
calc_shape_pal()

**Details**
This palette uses unicode symbols for the shapes. It will not work for all graphics devices.

**See Also**
Other shapes calc: `scale_shape_calc()`

**Examples**
```r
library("ggplot2")

## Not run:
show_shapes(calc_shape_pal()(13))

## End(Not run)
```
Canva.com color palettes

Description

150+ color palettes from canva.com. See `canva_palettes()`.

Usage

canva_pal(palette = "Fresh and bright")

Arguments

palette          Palette name. See the names of `canva_palettes()` for valid names.

Value

A function that takes a single value, the number of colors to use.

Examples

```r
require("ggplot2")
require("tibble")
if (require("purrr") && require("scales") && require("dplyr")) {
  canva_df <- map2_df(
    canva_palettes,
    names(canva_palettes),
    ~ tibble(
      colors = .x,
      .id = seq_along(colors),
      palette = .y
    )
  )
  ggplot(canva_df, aes(
    y = palette,
    x = .id,
    fill = colors
  )) +
  geom_raster() +
  scale_fill_identity(guide = FALSE) +
  theme_minimal() +
  theme(panel.grid = element_blank(),
    axis.text.x = element_blank()) +
  labs(x = "", y = "")

  show_col(canva_pal("Fresh and bright")(4))
  show_col(canva_pal("Cool blues")(4))
  show_col(canva_pal("Modern and crisp")(4))
}
```
150 Color Palettes from Canva

Description

150 four-color palettes by the canva.com design school. These palettes were derived from photos and "impactful websites".

Usage

canva_palettes

Format

A named list of character vector. The names are the palette names. The values of the character vectors are hex colors, e.g. "#f98866".

References

- Janie Kliever, 100 Brilliant Color Combinations and How to Apply Them to Your Designs, Canva.com, June 20, 2015.

Examples

```r
require("ggplot2")
require("tibble")
if (require("purrr") && require("scales") && require("dplyr")) {
  canva_df <- map2_df(
    canva_palettes,
    names(canva_palettes),
    ~ tibble(
      colors = .x,
      .id = seq_along(colors),
      palette = .y
    )
  )
  ggplot(canva_df, aes(
    y = palette,
    x = .id,
    fill = colors
  )) +
  geom_raster() +
  scale_fill_identity(guide = FALSE) +
  theme_minimal() +
  theme(panel.grid = element_blank(),
```
circlefill_shape_pal

```
axis.text.x = element_blank()) +
labs(x = "", y = "")

show_col(canva_pal("Fresh and bright")(4))
show_col(canva_pal("Cool blues")(4))
show_col(canva_pal("Modern and crisp")(4))
```

circlefill_shape_pal  Filled Circle Shape palette (discrete)

Description

'r lifecycle::badge("deprecated")'
This function was deprecated because unicode glyphs used for the circles vary in size, making them unusable for plotting.
Shape palette with circles varying by amount of fill. This uses the set of 3 circle fill values in Lewandowsky and Spence (1989): solid, hollow, half-filled, with two additional fill amounts: three-quarters, and one-quarter.
This palette supports up to five values.

Usage

circlefill_shape_pal()

References


See Also

Other shapes: cleveland_shape_pal(), scale_shape_circlefill(), scale_shape_cleveland(), scale_shape_tremmel(), tremmel_shape_pal()

cleveland_shape_pal  Shape palette from Cleveland "Elements of Graphing Data" (discrete).

Description

Shape palettes for overlapping and non-overlapping points.

Usage

cleveland_shape_pal(overlap = TRUE)
Arguments

overlap logical Use the scale for overlapping points?

Note

In the *Elements of Graphing Data*, W.S. Cleveland suggests two shape palettes for scatter plots: one for overlapping data and another for non-overlapping data. The symbols for overlapping data relies on pattern discrimination, while the symbols for non-overlapping data vary the amount of fill. This palette attempts to create these palettes. However, I found that these were hard to replicate. Using the R shapes and unicode fonts: the symbols can vary in size, they are dependent of the fonts used, and there does not exist a unicode symbol for a circle with a vertical line. If someone can improve this palette, please let me know.

Following Tremmel (1995), this function replaces the circle with a vertical line with an encircled plus sign.

The palette `cleveland_shape_pal()` supports up to five values.

This package uses unicode symbols for the shapes. This means that it will not work for all graphics devices.

References


See Also

Other shapes: `circlefill_shape_pal()`, `scale_shape_circlefill()`, `scale_shape_cleveland()`, `scale_shape_tremmel()`, `tremmel_shape_pal()`

Examples

```r
# Not run:
library("ggplot2")
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, shape = factor(gear))) +
  facet_wrap(~am) +
  theme_bw()
# overlapping symbol palette
p + scale_shape_cleveland()
# non-overlapping symbol palette
p + scale_shape_cleveland(overlap = FALSE)

# End(Not run)
```
Description

An eight-color colorblind safe qualitative discrete palette.

Usage

colorblind_pal()

scale_colour_colorblind(...)

scale_color_colorblind(...)

scale_fill_colorblind(...)

Arguments

Arguments passed on to discrete_scale

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).

breaks One of:
  • NULL for no breaks
  • waiver() for the default breaks (the scale limits)
  • A character vector of breaks
  • A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:
  • NULL to use the default scale values
  • A character vector that defines possible values of the scale and their order
  • A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.
labels

One of:

- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- An expression vector (must be the same length as `breaks`). See `plotmath` for details.
- A function that takes the breaks as input and returns labels as output.
  Also accepts rlang lambda function notation.

guide

A function used to create a guide or its name. See `guides()` for more information.

expand

For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function `expansion()` to generate the values for the `expand` argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position

For position scales, The position of the axis. `left` or `right` for y axes, `top` or `bottom` for x axes.

super

The super class to use for the constructed scale

References

Chang, W. "Cookbook for R"
https://jfly.iam.u-tokyo.ac.jp/color

See Also

The `dichromat` package, `dichromat_pal()`, and `scale_color_tableau()` for other colorblind palettes.

Examples

```r
library("ggplot2")
library("scales")

show_col(colorblind_pal()(8))
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg,
colour = factor(gear))) + facet_wrap(~am)
p + theme_igray() + scale_colour_colorblind()
```

economist_pal  

`Economist color palette (discrete)`

Description

The hues in the palette are blues, grays, and greens. Red is not included in these palettes and should be used to indicate important data.
`excel_new_pal`

Usage

```r
economist_pal(fill = TRUE)
```

Arguments

- `fill` Use the fill palette.

See Also

Other colour economist: `scale_colour_economist()`

Examples

```r
library("scales")

df <- data.frame(x = 1:10, y = rnorm(10))

show_col(economist_pal()(6))
## fill palette

show_col(economist_pal(fill = TRUE)(6))
```

---

**excel_new_pal**

*Excel (current versions) color palettes (discrete)*

**Description**

Color palettes used by current versions of Microsoft Office and Excel.

**Usage**

```r
eexcel_new_pal(theme = "Office Theme")
```

**Arguments**


**See Also**

Other colour excel: `excel_pal()`, `scale_colour_excel_new()`, `scale_fill_excel()`
Examples

```r
library("scales")
for (i in names(ggthemes::ggthemes_data$excel$palettes)) {
    show_col(excel_new_pal(theme = i))(6)
}
```

---

**excel_pal**  
Excel 97 ugly color palettes (discrete)

Description

The color palettes used in Microsoft Excel 97 (and up until Excel 2007). Use this for that classic ugly look and feel. For ironic purposes only. 3D bars and pies not included. Please never use this color palette.

Usage

```r
excel_pal(line = TRUE)
```

Arguments

- **line**
  
  If TRUE, use the palette for lines and points. Otherwise, use the palette for area.

See Also

Other colour excel: `excel_new_pal()`, `scale_colour_excel_new()`, `scale_fill_excel()`

Examples

```r
library("scales")
show_col(excel_pal()(7))
show_col(excel_pal(line = FALSE)(7))
```

---

**extended_range_breaks_**

Pretty axis breaks inclusive of extreme values

Description

This function returns pretty axis breaks that always include the extreme values of the data. This works by calling the extended Wilkinson algorithm (Talbot et. al, 2010), constrained to solutions interior to the data range. Then, the minimum and maximum labels are moved to the minimum and maximum of the data range.
Usage

```r
extended_range_breaks_(
  dmin,
  dmax,
  n = 5,
  Q = c(1, 5, 2, 2.5, 4, 3),
  w = c(0.25, 0.2, 0.5, 0.05)
)
```

extended_range_breaks(n = 5, ...)

Arguments

- `dmin`: minimum of the data range
- `dmax`: maximum of the data range
- `n`: desired number of breaks
- `Q`: set of nice numbers
- `w`: weights applied to the four optimization components (simplicity, coverage, density, and legibility)
- `...`: other arguments passed to `extended_range_breaks_()`

Details

`extended_range_breaks` implements the algorithm and returns the break values. `scales_extended_range_breaks` uses the conventions of the `scales` package, and returns a function.

Value

For `extended_range_breaks`, the vector of axis label locations. For `scales_extended_range_breaks`, a function which takes a single argument, a vector of data, and returns the vector of axis label locations.

A function which returns breaks given a vector.

Author(s)

Justin Talbot <jtalbot@stanford.edu>, Jeffrey B. Arnold, Baptiste Auguié

References

Description

Qualitative color palettes from Stephen Few (2012) Show Me the Numbers. There are three palettes: Light, Medium, and Dark. Each palette comprises nine colors: gray, blue, orange, green, pink, brown, purple, yellow, red. For \( n = 1 \), gray is used. For \( n > 1 \), the eight non-gray colors are used.

Usage

```r
few_pal(palette = "Medium")
```

Arguments

`palette` One of

Details

Use the light palette for filled areas, such as bar charts. Use the medium palette for points and lines. Use the dark palette for highlighting specific points or for small and thin lines and points.

References


"Practical Rules for Using Color in Charts".

See Also

Other colour few: `scale_colour_few()`

Examples

```r
library("scales")

show_col(few_pal()(7))
show_col(few_pal("Dark")(7))
show_col(few_pal("Light")(7))`
few_shape_pal  

Shape palette from "Show Me the Numbers" (discrete)

Description
Shape palette from Stephen Few's, "Show Me the Numbers". The shape palette consists of five shapes: circle, square, triangle, plus, times.

Usage
few_shape_pal()

References

fivethirtyeight_pal  

FiveThirtyEight color palette

Description
The standard three-color FiveThirtyEight palette for line plots comprises blue, red, and green.

Usage
fivethirtyeight_pal()

See Also
Other colour fivethirtyeight: scale_colour_fivethirtyeight()

Examples
library("scales")

show_col(fivethirtyeight_pal()(3))
gdocs_pal  Google Docs color palette (discrete)

Description

Color palettes from Google Docs. This palette includes 20 colors.

Usage

gdocs_pal()

See Also

Other colour gdocs: scale_fill_gdocs()

Examples

library("scales")

show_col(gdocs_pal()(20))

geom_rangeframe  Range Frames

Description

Axis lines which extend to the maximum and minimum of the plotted data.

Usage

geom_rangeframe(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ..., 
  sides = "bl",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
Arguments

mapping  Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.

data  The data to be displayed in this layer. There are three options:
If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.
A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.
A function will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A function can be created from a formula (e.g. `~ head(.x, 10)`).

stat  The statistical transformation to use on the data for this layer, either as a ggproto Geom subclass or as a string naming the stat stripped of the stat_ prefix (e.g. "count" rather than "stat_count")

position  Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use `position_jitter`), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.

...  Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

sides  A string that controls which sides of the plot the frames appear on. It can be set to a string containing any of ‘trbl’, for top, right, bottom, and left.

na.rm  If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

show.legend  logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.

inherit.aes  If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. `borders()`.

Details

This should be used with ‘coord_cartesian(clip="off")’ in order to correctly draw the lines.

Aesthetics

• colour
• size
• linetype
• alpha
References


See Also

Other geom tufte: `geom_tufteboxplot()`

Examples

```r
library("ggplot2")

ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_rangeframe() +
  coord_cartesian(clip="off") +
  theme_tufte()
```

# In the example above,
# `coord_cartesian(clip="off")` ensures that the full width of the line is drawn.
# if you know a better way to fix this,
# please open an issue or PR on github https://github.com/jrnold/ggthemes/issue

---

`geom_tufteboxplot`  
* Tufte’s Box Plot

Description

Edward Tufte’s revisions of the box plot as described in *The Visual Display of Quantitative Information*. This functions provides several box plot variants:

- A point indicating the median, a gap indicating the interquartile range, and lines for whiskers.
- An offset line indicating the interquartile range and a gap indicating the median.
- A line indicating the interquartile range, a gap indicating the median, and points indicating the minimum and maximum values
- A wide line indicating the interquartile range, a gap indicating the median, and lines indicating the minimum and maximum.

Usage

```r
geom_tufteboxplot(
  mapping = NULL,
  data = NULL,
  stat = "fivenumber",
  position = "dodge",
  outlier.colour = "black",
  outlier.shape = 19,
  outlier.size = 1.5,
)```
outlier.stroke = 0.5,
voffset = 0.01,
hoffset = 0.005,
na.rm = FALSE,
show.legend = NA,
inherit.aes = TRUE,
median.type = "point",
whisker.type = "line",
...
)

Arguments

mapping Set of aesthetic mappings created by \texttt{aes()}. If specified and \texttt{inherit.aes = TRUE} (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

data The data to be displayed in this layer. There are three options:
If \texttt{NULL}, the default, the data is inherited from the plot data as specified in the call to \texttt{ggplot()}. A \texttt{data.frame}, or other object, will override the plot data. All objects will be fortified to produce a data frame. See \texttt{fortify()} for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a \texttt{data.frame}, and will be used as the layer data. A function can be created from a \texttt{formula} (e.g. \texttt{~ head(.x, 10)}).

stat The statistical transformation to use on the data for this layer, either as a ggproto Geom subclass or as a string naming the stat stripped of the \texttt{stat_} prefix (e.g. \texttt{"count"} rather than \texttt{"stat_count"})

position Position adjustment, either as a string naming the adjustment (e.g. \texttt{"jitter"} to use \texttt{position_jitter}), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.

outlier.colour colour for outlying points
outlier.shape shape of outlying points
outlier.size size of outlying points
outlier.stroke stroke for outlying points
voffset controls the size of the gap in the line representing the median when \texttt{median.type = 'line'}. This is a fraction of the range of y.
hoffset controls how much the interquartile line is offset from the whiskers when \texttt{median.type = 'line'}. This is a fraction of the range of x.
na.rm If \texttt{FALSE}, the default, missing values are removed with a warning. If \texttt{TRUE}, missing values are silently removed.
show.legend logical. Should this layer be included in the legends? \texttt{NA}, the default, includes if any aesthetics are mapped. \texttt{FALSE} never includes, and \texttt{TRUE} always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. borders().

median.type If 'point', then the median is represented by a point, and the interquartile range by a gap in the line. If median.type='line', then the interquartile range is represented by a line, possibly offset, and the median by a gap in the line.

whisker.type If 'line', then whiskers are represented by lines. If 'point', then whiskers are represented by points at ymin and ymax.

... Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.

Aesthetics

• x [required]
• y [required]
• colour
• size
• linetype
• shape
• fill
• alpha

References


See Also

geom_boxplot() Other geom tufte: geom_rangeframe()

Examples

library("ggplot2")

p <- ggplot(mtcars, aes(factor(cyl), mpg))
# with a point for the median and lines for whiskers
p + geom_tufteboxplot()
# with a line for the interquartile range and points for whiskers
p + geom_tufteboxplot(median.type = "line", whisker.type = "point", hoffset = 0)
# with a wide line for the interquartile range and lines for whiskers
p + geom_tufteboxplot(median.type = "line", hoffset = 0, width = 3)
# with an offset line for the interquartile range and lines for whiskers
p + geom_tufteboxplot(median.type = "line")
# combined with theme_tufte
p + geom_tufteboxplot() +
theme_tufte() +
theme(axis.ticks.x = element_blank())

---

**ggthemes**

**Description**

This package contains extra themes, scales, and geoms, and functions for and related to **ggplot2**. See [https://jrnold.github.io/ggthemes/](https://jrnold.github.io/ggthemes/) for documentation.

**Author(s)**

**Maintainer**: Jeffrey B. Arnold &lt;jeffrey.arnold@gmail.com&gt; (&lt;https://orcid.org/0000-0001-9953-3904&gt;)

Other contributors:

- Gergely Daroczi [contributor]
- Bo Werth [contributor]
- Brian Weitzner [contributor]
- Joshua Kunst [contributor]
- Baptise Auguie [contributor]
- Bob Rudis [contributor]
- Hadley Wickham (Code from the ggplot2 package.) [contributor]
- Justin Talbot (Code from the labeling package) [contributor]
- Joshua London [contributor]

**See Also**

Useful links:

- [https://github.com/jrnold/ggthemes](https://github.com/jrnold/ggthemes)
- Report bugs at [https://github.com/jrnold/ggthemes](https://github.com/jrnold/ggthemes)
ggthemes_data | Palette and theme data

Description

The ggthemes environment contains various values used in themes and palettes. This is undocumented and subject to change.

Usage

ggthemes_data

Format

A list object.

hc_pal | Highcharts color palette (discrete)

Description

The Highcharts uses many different color palettes in its plots. This collects a few of them.

Usage

hc_pal(palette = "default")

Arguments

data character The name of the Highcharts theme to use.

Palettes

The following palettes are defined,

- default
- dark-unica

See Also

Other colour hc: scale_colour_hc()
palette_pander

Color palette from the pander package

Description

The pander ships with a default colorblind and printer-friendly color palette borrowed from https://jfly.iam.u-tokyo.ac.jp/color/

Usage

palette_pander(n, random_order = FALSE)

Arguments

n number of colors. This palette supports up to eight colors.
random_order if the palette should be reordered randomly before rendering each plot to get colorful images

See Also

Other colour pander: scale_color_pander()

Examples

## Not run:
    palette_pander(TRUE)

## End(Not run)

ptol_pal

Color Palettes from Paul Tol’s “Colour Schemes”

Description

Qualitative color palettes from Paul Tol, "Colour Schemes".

Usage

ptol_pal()

Details

Incorporation of the palette into an R package was originally inspired by Peter Carl’s [Paul Tol 21 Gun Salute](https://tradeblotter.wordpress.com/2013/02/28/the-paul-tol-21-color-salute/)
References


See Also

Other colour ptol: scale_colour_ptol()

Examples

library("scales")

show_col(ptol_pal()(6))
show_col(ptol_pal()(4))
show_col(ptol_pal()(12))

scale_color_pander \hspace{1cm} \textit{Color scale from the pander package}

Description

The \texttt{pander} ships with a default colorblind and printer-friendly color palette borrowed from https://jfly.iam.u-tokyo.ac.jp/color/.

Usage

scale_color_pander(...)  
scale_colour_pander(...)  
scale_fill_pander(...)

Arguments

...  
\hspace{1cm} \textit{Arguments passed on to \texttt{discrete_scale}}

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., \texttt{scales::hue_pal()}).

breaks One of:
- \texttt{NULL} for no breaks
- \texttt{wavier()} for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang \texttt{lambda} function notation.

limits One of:
- \texttt{NULL} to use the default scale values
• A character vector that defines possible values of the scale and their order
• A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels One of:
• NULL for no labels
• waiver() for the default labels computed by the transformation object
• A character vector giving labels (must be same length as breaks)
• An expression vector (must be the same length as breaks). See ?plotmath for details.
• A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also

theme_pander()

Other colour pander: palette_pander()

scale_colour_canva Discrete color scale using canva.com color palettes

Description

Color scale for canva.com color palettes described in canva_palettes().
scale_colour_economist

Economist color scales

Description

Color scales using the colors in the Economist graphics.

Usage

scale_colour_economist(...)

scale_color_economist(...)

scale_fill_economist(...)

Arguments

... Arguments passed on to discrete_scale.

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).

breaks One of:
  • NULL for no breaks
  • waiver() for the default breaks (the scale limits)
  • A character vector of breaks
  • A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:
  • NULL to use the default scale values
  • A character vector that defines possible values of the scale and their order
**scale_colour_excel_new**

- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

**drop**  Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

**na.translate**  Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

**scale_name**  The name of the scale that should be used for error messages associated with this scale.

**name**  The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

**labels**  One of:
- NULL for no labels
- waiver() for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

**guide**  A function used to create a guide or its name. See guides() for more information.

**expand**  For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

**position**  For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

**super**  The super class to use for the constructed scale

---

**See Also**

theme_economist() for examples.

Other colour economist: economist_pal()
Usage

scale_colour_excel_new(theme = "Office Theme", ...)
scale_color_excel_new(theme = "Office Theme", ...)
scale_fill_excel_new(theme = "Office Theme", ...)

Arguments


Arguments passed on to discrete_scale

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).
breaks One of:
- NULL for no breaks
- waiver() for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:
- NULL to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.
drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.
na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.
scale_name The name of the scale that should be used for error messages associated with this scale.
name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.
labels One of:
• NULL for no labels
• waiver() for the default labels computed by the transformation object
• A character vector giving labels (must be the same length as breaks)
• An expression vector (must be the same length as breaks). See ?plotmath for details.
• A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also
Other colour excel: excel_new_pal(), excel_pal(), scale_fill_excel()

Examples

library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)

p + theme_excel_new() + scale_colour_excel_new()
Arguments

Arguments passed on to `discrete_scale`

- **palette**
  A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

- **breaks**
  One of:
  - `NULL` for no breaks
  - `waiver()` for the default breaks (the scale limits)
  - A character vector of breaks
  - A function that takes the limits as input and returns breaks as output.
    Also accepts rlang lambda function notation.

- **limits**
  One of:
  - `NULL` to use the default scale values
  - A character vector that defines possible values of the scale and their order
  - A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

- **drop**
  Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

- **na.translate**
  Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

- **scale_name**
  The name of the scale that should be used for error messages associated with this scale.

- **name**
  The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

- **labels**
  One of:
  - `NULL` for no labels
  - `waiver()` for the default labels computed by the transformation object
  - A character vector giving labels (must be same length as breaks)
  - An expression vector (must be the same length as breaks). See `plotmath` for details.
  - A function that takes the breaks as input and returns labels as output.
    Also accepts rlang lambda function notation.

- **guide**
  A function used to create a guide or its name. See `guides()` for more information.

- **expand**
  For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function `expansion()` to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.
position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

See Also

Other colour few: few_pal()

Description

Color scales using the colors in the FiveThirtyEight graphics.

Usage

scale_colour_fivethirtyeight(...)
scale_color_fivethirtyeight(...)
scale_fill_fivethirtyeight(...)

Arguments

... Arguments passed on to discrete_scale

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).

breaks One of:

• NULL for no breaks
• waiver() for the default breaks (the scale limits)
• A character vector of breaks
• A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:

• NULL to use the default scale values
• A character vector that defines possible values of the scale and their order
• A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.
scale_colour_gradient2_tableau

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels One of:
- NULL for no labels
- waiver() for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also

theme_fivethirtyeight() for examples.

Other colour fivethirtyeight: fivethirtyeight_pal()

scale_colour_gradient2_tableau

Tableau diverging colour scales (continuous)

Description

Continuous color scales using the diverging color scales in Tableau. See scale_colour_tableau() for Tabaleau discrete color scales, and scale_colour_gradient_tableau() for sequential color scales.
scale_colour_gradient2_tableau

Usage

scale_colour_gradient2_tableau(
  palette = "Orange-Blue Diverging",
  ..., 
  na.value = "grey50",
  guide = "colourbar"
)

table = "Orange-Blue Diverging",
  ..., 
  na.value = "grey50",
  guide = "colourbar"
)

table = "Orange-Blue Diverging",
  ..., 
  na.value = "grey50",
  guide = "colourbar"
)

Arguments

  palette  Palette name. One of the following:

  Ordered sequential palettes "Blue-Green Sequential", "Blue Light", "Orange
"Gray Warm", "Blue-Teal", "Orange-Gold", "Green-Gold", "Red-Gold",
Orange", "Classic Area Red", "Classic Area Green", "Classic Area-Brown"

  Ordered diverging palettes "Orange-Blue Diverging", "Red-Green Diverging",
"Green-Blue Diverging", "Red-Blue Diverging", "Red-Black Diverging",
"Gold-Purple Diverging", "Red-Green-Gold Diverging", "Sunset-Sunrise
Diverging", "Orange-Blue-White Diverging", "Red-Green-White Diverging",
"Green-Blue-White Diverging", "Red-Blue-White Diverging", "Red-Black-White
Diverging", "Orange-Blue Light Diverging", "Temperature Diverging",
"Classic Orange-Green", "Classic Orange-Blue", "Classic Orange-Black"

  ...  Arguments passed to tableau_gradient_pal.

  na.value  Colour to use for missing values

  guide  Type of legend. Use 'colourbar' for continuous colour bar, or 'legend' for
discrete colour legend.
See Also

Other colour tableau: `scale_colour_gradient_tableau()`, `scale_colour_tableau()`, `tableau_color_pal()`, `tableau_gradient_pal()`

Examples

```r
library("ggplot2")

df <- data.frame(
  x = runif(100),
  y = runif(100),
  z1 = rnorm(100),
  z2 = abs(rnorm(100))
)
p <- ggplot(df, aes(x, y)) + geom_point(aes(colour = z2))

palettes <-
  ggthemes_data["tableau"]["color-palettes"][["ordered-diverging"]]
for (palette in head(names(palettes))) {
  print(p + scale_colour_gradient2_tableau(palette) + ggtitle(palette))
}

# If you need to reverse a palette, use a transformation
p + scale_colour_gradient2_tableau(trans = "reverse")
```

---

`scale_colour_gradient_tableau`

Tableau sequential colour scales (continuous)

Description

Continuous color scales using the sequential color palettes in Tableau. See `scale_colour_tableau()` for Tableau discrete color scales, and `scale_colour_gradient2_tableau()` for diverging color scales.

Usage

```r
scale_colour_gradient_tableau(
  palette = "Blue",
  ...,
  na.value = "grey50",
  guide = "colourbar"
)

scale_fill_gradient_tableau(
  palette = "Blue",
  ...,
```
scale_colour_gradient_tableau

    na.value = "grey50",
    guide = "colourbar"
)

scale_color_gradient_tableau(
    palette = "Blue",
    ...
    na.value = "grey50",
    guide = "colourbar"
)

scale_color_continuous_tableau(
    palette = "Blue",
    ...
    na.value = "grey50",
    guide = "colourbar"
)

scale_fill_continuous_tableau(
    palette = "Blue",
    ...
    na.value = "grey50",
    guide = "colourbar"
)

Arguments

palette          Palette name. One of the following:

    Ordered sequential palettes  "Blue-Green Sequential", "Blue Light", "Orange
"Gray Warm", "Blue-Teal", "Orange-Gold", "Green-Gold", "Red-Gold",
Orange", "Classic Area Red", "Classic Area Green", "Classic Area-Brown"

    Ordered diverging palettes  "Orange-Blue Diverging", "Red-Green Diverging",
"Green-Blue Diverging", "Red-Blue Diverging", "Red-Black Diverging",
"Gold-Purple Diverging", "Red-Green-Gold Diverging", "Sunset-Sunrise
Diverging", "Orange-Blue-White Diverging", "Red-Green-White Diverging",
"Green-Blue-White Diverging", "Red-Blue-White Diverging", "Red-Black-White
Diverging", "Orange-Blue Light Diverging", "Temperature Diverging",
"Classic Red-White-Black Light", "Classic Orange-White-Blue Light",
"Classic Red-White-Green Light", "Classic Red-Green Light"


...          Arguments passed to tableau_gradient_pal.

     na.value       Colour to use for missing values

guide          Type of legend. Use 'colourbar' for continuous colour bar, or 'legend'
for discrete colour legend.
See Also

Other colour tableau: `scale_colour_gradient2_tableau()`, `scale_colour_tableau()`, `tableau_color_pal()`, `tableau_gradient_pal()`

Examples

```r
library("ggplot2")

df <- data.frame(
  x = runif(100),
  y = runif(100),
  z1 = rnorm(100),
  z2 = abs(rnorm(100))
)

p <- ggplot(df, aes(x, y)) +
  geom_point(aes(colour = z2)) +
  theme_igray()

palettes <-
  ggthemes_data[["tableau"]][["color-palettes"]][["ordered-sequential"]]
for (palette in head(names(palettes))) {
  print(p + scale_colour_gradient_tableau(palette) +
        ggtitle(palette))
}
```

---

### scale_colour_hc  
**Highcharts color and fill scales**

#### Description

Colour and fill scales which use the palettes in `hc_pal()` and are meant for use with `theme_hc()`.

#### Usage

```r
scale_colour_hc(palette = "default", ...)
scale_color_hc(palette = "default", ...)
scale_fill_hc(palette = "default", ...)
```

#### Arguments

- **palette** character The name of the Highcharts theme to use.
- **...** Arguments passed on to `discrete_scale`
- **palette** A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).
breaks One of:
- NULL for no breaks
- waiver() for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:
- NULL to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels One of:
- NULL for no labels
- waiver() for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also
Other colour hc: hc_pal()
scale_colour_ptol

Color Scales from Paul Tol’s "Colour Schemes"

Description

See `ptol_pal()`. These palettes support up to 12 values.

Usage

```r
scale_colour_ptol(...)
scale_color_ptol(...)
scale_fill_ptol(...)```

Arguments

... Arguments passed on to `discrete_scale`

`palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

`breaks` One of:
- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

`limits` One of:
- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

`drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

`na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

`scale_name` The name of the scale that should be used for error messages associated with this scale.

`name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

`labels` One of:
scale_colour_stata

- NULL for no labels
- waiver() for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also

Other colour ptol: ptol_pal()

Examples

library("ggplot2")

p2 <- ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(gear))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_ptol("cyl") +
  theme_minimal() +
  ggtitle("Cars")

ggplot(diamonds, aes(x = clarity, fill = cut)) +
  geom_bar() +
  scale_fill_ptol() +
  theme_minimal()
scale_colour_stata

Usage

scale_colour_stata(scheme = "s2color", ...)  
scale_fill_stata(scheme = "s2color", ...)  
scale_color_stata(scheme = "s2color", ...)

Arguments

scheme character. One of "s2color", "s1rcolor", "s1color", or "mono".

... Arguments passed on to discrete_scale

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).

breaks One of:
- NULL for no breaks
- waiver() for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:
- NULL to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels One of:
- NULL for no labels
- waiver() for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.
guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

scale_colour_tableau  Tableau color scales (discrete)

Description

Categorical (qualitative) color scales used in Tableau. Use the function scale_colour_gradient_tableau() for the sequential and scale_colour_gradient2_tableau() for the diverging continuous color scales from Tableu.

Usage

scale_colour_tableau(
   palette = "Tableau 10",
   type = "regular",
   direction = 1,
   ...
)

scale_fill_tableau(
   palette = "Tableau 10",
   type = "regular",
   direction = 1,
   ...
)

scale_color_tableau(
   palette = "Tableau 10",
   type = "regular",
   direction = 1,
   ...
)
scale_colour_wsj

Arguments

- `palette` (Palette name. See `tableau_color_pal()` for available palettes.)
- `type` (Palette type. One of "regular", "sequential", or "diverging". See `tableau_color_pal()`.)
- `direction` (If 1, the default, then use the original order of colors. If -1, then reverse the order.)
- `...` (Other arguments passed on to `discrete_scale()`.)

See Also

- `tableau_color_pal()` for references.
- Other colour tableau: `scale_colour_gradient2_tableau()`, `scale_colour_gradient_tableau()`, `tableau_color_pal()`, `tableau_gradient_pal()`

Examples

```r
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am) +
  theme_igray()

palettes <- ggthemes_data["tableau"][["color-palettes"]][["regular"]]
for (palette in head(names(palettes), 3L)) {
  print(p + scale_colour_tableau(palette) +
        ggtitle(palette))
}

# the order of colour can be reversed
p + scale_color_tableau(direction = -1)
```

---

scale_colour_wsj  
*Wall Street Journal color and fill scales*

Description

Colour and fill scales which use the palettes in `wsj_pal()`. These scales should be used with `theme_wsj()`.

Usage

- `scale_colour_wsj(palette = "colors6", ...)`
- `scale_color_wsj(palette = "colors6", ...)`
- `scale_fill_wsj(palette = "colors6", ...)`
Arguments

palette  character The color palette to use: "rgby", "red_green", "black_green", "dem_rep", "colors6"

...  Arguments passed on to discrete_scale

palette  A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).

breaks  One of:
  • NULL for no breaks
  • waiver() for the default breaks (the scale limits)
  • A character vector of breaks
  • A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits  One of:
  • NULL to use the default scale values
  • A character vector that defines possible values of the scale and their order
  • A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop  Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate  Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

ccale_name  The name of the scale that should be used for error messages associated with this scale.

name  The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels  One of:
  • NULL for no labels
  • waiver() for the default labels computed by the transformation object
  • A character vector giving labels (must be same length as breaks)
  • An expression vector (must be the same length as breaks). See ?plotmath for details.
  • A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide  A function used to create a guide or its name. See guides() for more information.

expand  For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.
position For position scales, the position of the axis. left or right for y axes, top or bottom for x axes.
super The super class to use for the constructed scale

See Also
Other colour wsj: wsj_pal()

scale_fill_calc LibreOffice Calc color scales

Description
Color scales from LibreOffice Calc.

Usage
scale_fill_calc(...) scale_colour_calc(...) scale_color_calc(...)

Arguments
... Arguments passed on to discrete_scale
palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).
breaks One of:
  • NULL for no breaks
  • waiver() for the default breaks (the scale limits)
  • A character vector of breaks
  • A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.
limits One of:
  • NULL to use the default scale values
  • A character vector that defines possible values of the scale and their order
  • A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.
drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.
na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.
scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels One of:
  - NULL for no labels
  - waiver() for the default labels computed by the transformation object
  - A character vector giving labels (must be same length as breaks)
  - An expression vector (must be the same length as breaks). See ?plotmath for details.
  - A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also
See theme_calc() for examples.
Other colour calc: calc_pal()

scale_fill_excel Excel 97 ugly color scales

Description
The classic "ugly" color scales from Excel 97.

Usage
scale_fill_excel(...)  
scale_colour_excel(...)  
scale_color_excel(...)
**Arguments**

Arguments passed on to `discrete_scale`

- **palette** A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

- **breaks** One of:
  - `NULL` for no breaks
  - `waiver()` for the default breaks (the scale limits)
  - A character vector of breaks
  - A function that takes the limits as input and returns breaks as output. Also accepts rlang `lambda` function notation.

- **limits** One of:
  - `NULL` to use the default scale values
  - A character vector that defines possible values of the scale and their order
  - A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang `lambda` function notation.

- **drop** Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

- **na.translate** Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

- **scale_name** The name of the scale that should be used for error messages associated with this scale.

- **name** The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

- **labels** One of:
  - `NULL` for no labels
  - `waiver()` for the default labels computed by the transformation object
  - A character vector giving labels (must be same length as `breaks`)
  - An expression vector (must be the same length as `breaks`). See `?plotmath` for details.
  - A function that takes the breaks as input and returns labels as output. Also accepts rlang `lambda` function notation.

- **guide** A function used to create a guide or its name. See `guides()` for more information.

- **expand** For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function `expansion()` to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

- **position** For position scales, the position of the axis. `left` or `right` for y axes, `top` or `bottom` for x axes.

- **super** The super class to use for the constructed scale
See Also

Other colour excel: `excel_new_pal()`, `excel_pal()`, `scale_colour_excel_new()`

Examples

```r
library("ggplot2")

# Line and scatter plot colors
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)

p + theme_excel() + scale_colour_excel()

# Bar plot (area/fill) colors
ggplot(mpg, aes(x = class, fill = drv)) +
  geom_bar() +
  scale_fill_excel() +
  theme_excel()
```

---

**scale_fill_gdocs**

Google Docs color scales

Description

Color scales from Google Docs.

Usage

```r
scale_fill_gdocs(...)

scale_colour_gdocs(...)

scale_color_gdocs(...)
```

Arguments

... Arguments passed on to `discrete_scale`

- `palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

- `breaks` One of:
  - NULL for no breaks
  - `waiver()` for the default breaks (the scale limits)
  - A character vector of breaks
  - A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

- `limits` One of:
• NULL to use the default scale values
• A character vector that defines possible values of the scale and their order
• A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.

labels One of:
• NULL for no labels
• waiver() for the default labels computed by the transformation object
• A character vector giving labels (must be same length as breaks)
• An expression vector (must be the same length as breaks). See plotmath for details.
• A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

guide A function used to create a guide or its name. See guides() for more information.

expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.

position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.

super The super class to use for the constructed scale

See Also

See theme_gdocs() for examples.

Other colour gdocs: gdocs_pal()
**scale_fill_solarized**  *Solarized color scales*

**Description**

See `solarized_pal()` for details.

**Usage**

```r
scale_fill_solarized(accent = "blue", ...)
scale_colour_solarized(accent = "blue", ...)
scale_color_solarized(accent = "blue", ...)
```

**Arguments**

- `accent` character Starting color.
- `...` Arguments passed on to `discrete_scale`
- `palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).
- `breaks` One of:
  - `NULL` for no breaks
  - `waiver()` for the default breaks (the scale limits)
  - A character vector of breaks
  - A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.
- `limits` One of:
  - `NULL` to use the default scale values
  - A character vector that defines possible values of the scale and their order
  - A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.
- `drop` Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.
- `na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.
- `scale_name` The name of the scale that should be used for error messages associated with this scale.
- `name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.
labels One of:
- NULL for no labels
- waiver() for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output.
  Also accepts rlang lambda function notation.
guide A function used to create a guide or its name. See guides() for more information.
expand For position scales, a vector of range expansion constants used to add some padding around the data to ensure that they are placed some distance away from the axes. Use the convenience function expansion() to generate the values for the expand argument. The defaults are to expand the scale by 5% on each side for continuous variables, and by 0.6 units on each side for discrete variables.
position For position scales, The position of the axis. left or right for y axes, top or bottom for x axes.
super The super class to use for the constructed scale

See Also
Other solarized colour: solarized_pal()

Examples
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)
p + theme_solarized() +
scale_colour_solarized()
Arguments

Arguments passed on to `discrete_scale`

apalette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

breaks One of:
- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

limits One of:
- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

na.value If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where NA is always placed at the far right.

aesthetics The names of the aesthetics that this scale works with.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

labels One of:
- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as breaks)
- An expression vector (must be the same length as breaks). See ?plotmath for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

super The super class to use for the constructed scale

See Also

Other linetype stata: `stata_linetype_pal()`
Examples

```r
require("ggplot2")
if (require("tidyr") & require("dplyr")) {
  rescale01 <- function(x) {
    (x - min(x)) / diff(range(x))
  }

gather(economics, variable, value, -date) %>%
  group_by(variable) %>%
  mutate(value = rescale01(value)) %>%
  ggplot(aes(x = date, y = value, linetype = variable)) +
  geom_line() +
  scale_linetype_stata()
}
```

scale_shape_calc  Calc shape scale

Description

See `calc_shape_pal()` for details.

Usage

```r
scale_shape_calc(...)
```

Arguments

`...` Arguments passed on to `discrete_scale`

`palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

`breaks` One of:

- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang `lambda` function notation.

`limits` One of:

- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang `lambda` function notation.

`drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.
Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

`na.value` If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where `NA` is always placed at the far right.

`aesthetics` The names of the aesthetics that this scale works with.

`scale_name` The name of the scale that should be used for error messages associated with this scale.

`name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

`labels` One of:
- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- An expression vector (must be the same length as `breaks`). See `plotmath` for details.
- A function that takes the breaks as input and returns labels as output. Also accepts `rlang` `lambda` function notation.

`super` The super class to use for the constructed scale

### Details
This scale uses unicode symbols for the shapes. It will not work for all graphics devices.

### See Also
- `theme_calc()` for examples.

Other shapes calc: `calc_shape_pal()`

---

**scale_shape_circlefill**

*Filled Circle Shape palette (discrete)*

### Description

'\r lifecycle::badge("deprecated")'

### Usage

`scale_shape_circlefill(...)`
Arguments

Arguments passed on to `discrete_scale`

palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

breaks One of:
- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang `lambda` function notation.

limits One of:
- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang `lambda` function notation.

drop Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

na.value If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where `NA` is always placed at the far right.

aesthetics The names of the aesthetics that this scale works with.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

labels One of:
- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- An expression vector (must be the same length as `breaks`). See `plotmath` for details.
- A function that takes the breaks as input and returns labels as output. Also accepts rlang `lambda` function notation.

super The super class to use for the constructed scale

See Also

circlefill_shape_pal() for a description of the palette.
scale_shape_cleveland

Other shapes: circlefill_shape_pal(), cleveland_shape_pal(), scale_shape_cleveland(), scale_shape_tremmel(), tremmel_shape_pal()

scale_shape_cleveland Shape scales from Cleveland "Elements of Graphing Data"

Description

Shape scales from Cleveland "Elements of Graphing Data"

Usage

scale_shape_cleveland(overlap = TRUE, ...)

Arguments

overlap logical Use the scale for overlapping points?
...
Arguments passed on to discrete_scale
palette A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., scales::hue_pal()).
breaks One of:
  • NULL for no breaks
  • waiver() for the default breaks (the scale limits)
  • A character vector of breaks
  • A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.
limits One of:
  • NULL to use the default scale values
  • A character vector that defines possible values of the scale and their order
  • A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.
drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.
a.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify na.translate = FALSE.
na.value If na.translate = TRUE, what aesthetic value should the missing values be displayed as? Does not apply to position scales where NA is always placed at the far right.
aesthetics The names of the aesthetics that this scale works with.
scale_name The name of the scale that should be used for error messages associated with this scale.
name  The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

labels One of:
  • `NULL` for no labels
  • `waiver()` for the default labels computed by the transformation object
  • A character vector giving labels (must be same length as `breaks`)
  • An expression vector (must be the same length as breaks). See ?plotmath for details.
  • A function that takes the breaks as input and returns labels as output.
    Also accepts rlang `lambda` function notation.

super  The super class to use for the constructed scale

References

See Also
`cleveland_shape_pal()` for a description of the palette.

Other shapes: `circlefill_shape_pal()`, `cleveland_shape_pal()`, `scale_shape_circlefill()`, `scale_shape_tremmel()`, `tremmel_shape_pal()`

scale_shape_few  Scales for shapes from "Show Me the Numbers"

Description
`scale_shape_few()` maps discrete variables to up to five easily discernible shapes. It is based on the shape palette suggested in Few (2012).

Usage
`scale_shape_few(...)`

Arguments
...  Common `discrete_scale()` parameters.

References

See Also
`scale_shape_few()` for the shape palette that this scale uses.
scale_shape_stata  

**Stata shape scale**

**Description**

See `stata_shape_pal()` for details.

**Usage**

`scale_shape_stata(...)`

**Arguments**

`...`  
Arguments passed on to `discrete_scale`

`palette`  
A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

`breaks`  
One of:
- `NULL` for no breaks
- `waiver()` for the default breaks (the scale limits)
- A character vector of breaks
- A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

`limits`  
One of:
- `NULL` to use the default scale values
- A character vector that defines possible values of the scale and their order
- A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

`drop`  
Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE uses all the levels in the factor.

`na.translate`  
Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

`na.value`  
If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where NA is always placed at the far right.

`aesthetics`  
The names of the aesthetics that this scale works with.

`scale_name`  
The name of the scale that should be used for error messages associated with this scale.

`name`  
The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

`labels`  
One of:
scale_shape_tableau

• NULL for no labels
• waiver() for the default labels computed by the transformation object
• A character vector giving labels (must be same length as breaks)
• An expression vector (must be the same length as breaks). See ?plotmath for details.
• A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.

super The super class to use for the constructed scale

Details

This scale uses unicode symbols for the shapes. This means that it will not work for all graphics devices.

Examples

```r
## Not run:
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, shape = factor(gear))) +
  facet_wrap(~am)

p + theme_stata() + scale_shape_stata()

## End(Not run)
```

scale_shape_tableau Tableau shape scales

Description

See `tableau_shape_pal()` for details.

Usage

```r
scale_shape_tableau(palette = "default", ...)
```

Arguments

- `palette` Palette name.
- `...` Arguments passed on to `discrete_scale`
- `palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).
- `breaks` One of:
  - NULL for no breaks
• `waiver()` for the default breaks (the scale limits)
• A character vector of breaks
• A function that takes the limits as input and returns breaks as output. Also accepts rlang `lambda` function notation.

limits One of:
• `NULL` to use the default scale values
• A character vector that defines possible values of the scale and their order
• A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang `lambda` function notation.

drop Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

na.translate Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.

na.value If `na.translate = TRUE`, what aesthetic value should the missing values be displayed as? Does not apply to position scales where `NA` is always placed at the far right.

aesthetics The names of the aesthetics that this scale works with.

scale_name The name of the scale that should be used for error messages associated with this scale.

name The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.

labels One of:
• `NULL` for no labels
• `waiver()` for the default labels computed by the transformation object
• A character vector giving labels (must be same length as `breaks`)
• An expression vector (must be the same length as `breaks`). See `?plotmath` for details.
• A function that takes the breaks as input and returns labels as output. Also accepts rlang `lambda` function notation.

super The super class to use for the constructed scale

Details

This palette uses unicode symbols. It may not work on all graphics devices.

See Also

Other shape tableau: `tableau_shape_pal()`

Examples

```
## Not run:
library("ggplot2")
```
```r
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg, shape = factor(gear))) + facet_wrap(~am)
p + scale_shape_tableau()

## End(Not run)
```

**scale_shape_tremmel**  
*Shape scales from Tremmel (1995)*

**Description**
Shape scales from Tremmel (1995)

**Usage**
```r
scale_shape_tremmel(overlap = FALSE, alt = TRUE, ...)
```

**Arguments**
- `overlap` use an empty circle instead of a solid circle when \( n = 2 \).
- `alt` If `TRUE`, then when \( n = 3 \), use a solid circle, plus sign and empty triangle. Otherwise use a solid circle, empty circle, and empty triangle.
- `...` Arguments passed on to `discrete_scale`

- `palette` A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., `scales::hue_pal()`).

- `breaks` One of:
  - `NULL` for no breaks
  - `waiver()` for the default breaks (the scale limits)
  - A character vector of breaks
  - A function that takes the limits as input and returns breaks as output. Also accepts rlang lambda function notation.

- `limits` One of:
  - `NULL` to use the default scale values
  - A character vector that defines possible values of the scale and their order
  - A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation.

- `drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.

- `na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`. 

show_linetypes

na.value If na.translate = TRUE, what aesthetic value should the missing values be displayed as? Does not apply to position scales where NA is always placed at the far right.
aesthetics The names of the aesthetics that this scale works with.
scale_name The name of the scale that should be used for error messages associated with this scale.
name The name of the scale. Used as the axis or legend title. If waiver(), the default, the name of the scale is taken from the first mapping used for that aesthetic. If NULL, the legend title will be omitted.
labels One of:
  • NULL for no labels
  • waiver() for the default labels computed by the transformation object
  • A character vector giving labels (must be same length as breaks)
  • An expression vector (must be the same length as breaks). See ?plotmath for details.
  • A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation.
super The super class to use for the constructed scale

See Also
tremmel_shape_pal() for a description of the palette.
Other shapes: circlefill_shape_pal(), cleveland_shape_pal(), scale_shape_circlefill(), scale_shape_cleveland(), tremmel_shape_pal()

Examples

```r
## Not run:
library("ggplot2")

p <- ggplot(mtcars, aes(x = mpg, y = hp, shape = factor(cyl))) + geom_point()
p + scale_shape_tremmel()
p + scale_shape_tremmel(alt = TRUE)
p + scale_shape_tremmel(overlap = TRUE)
## End(Not run)
```

---

**Description**

A quick and dirty way to show linetypes.
**Usage**

```r
show_linetypes(linetypes, labels = TRUE)
```

**Arguments**

- `linetypes`: A character vector of linetypes. See `par()`.
- `labels`: Label each line with its linetype (lty) value.

**Value**

This function called for the side effect of creating a plot. It returns `linetypes`.

**See Also**

`show_col()`, `show_linetypes()`

**Examples**

```r
library("scales")

show_linetypes(linetype_pal()(3))
show_linetypes(linetype_pal()(3), labels = TRUE)
```

---

**show_shapes**

**Show shapes**

**Description**

A quick and dirty way to show shapes.

**Usage**

```r
show_shapes(shapes, labels = TRUE)
```

**Arguments**

- `shapes`: A numeric or character vector of shapes. See `par()`.
- `labels`: Include the plotting character value of the symbol.

**Value**

This function called for the side effect of creating a plot. It returns `shapes`.

**See Also**

`show_col()`, `show_linetypes()`
Examples

```r
library("scales")

show_shapes(shape_pal()(5))
show_shapes(shape_pal()(3), labels = TRUE)
```

---

**smart_digits**

*Format numbers with automatic number of digits*

**Description**

Format numbers with automatic number of digits

**Usage**

```r
smart_digits(x, ...)
smart_digits_format(x, ...)
```

**Arguments**

- `x` A numeric vector to format
- `...` Parameters passed to `format()`

**Value**

A character vector. `smart_digits_format()` returns a function with a single argument `x`, a numeric vector, that returns a character vector.

**Author(s)**

Josh O'Brien, Baptise Auguie, Jeffrey B. Arnold

**References**

solarized_pal  Solarized color palette (discrete)

Description
Qualitative color palette based on the Ethan Schoonover’s Solarized palette, https://ethanschoonover.com/solarized/. This palette supports up to seven values.

Usage
solarized_pal(accent = "blue")

Arguments
accent character Starting color.

Note
For a given starting color and number of colors in the palette, the other colors are the combination of colors that maximizes the total Euclidean distance between colors in L*a*b space.

See Also
Other solarized color: scale_fill_solarized()

Examples
library("scales")

show_col(solarized_pal()(2))
show_col(solarized_pal()(3))
show_col(solarized_pal("red")(4))

 stata_linetype_pal  Stata linetype palette (discrete)

Description
Linetype palette based on the linepattern scheme in Stata. This palette supports up to 15 values.

Usage
stata_linetype_pal()

See Also
scale_linetype_stata()
Other linetype stata: scale_linetype_stata()
Stata color palettes (discrete)

Description


Usage

stata_pal(scheme = "s2color")

Arguments

scheme character. One of "s2color", "s1rcolor", "s1color", or "mono".

Details

All these palettes support up to 15 values.

Examples

library("scales")

show_col(stata_pal("s2color")(15))
show_col(stata_pal("s1rcolor")(15))
show_col(stata_pal("s1color")(15))
show_col(stata_pal("mono")(15))

Stata shape palette (discrete)

Description

Shape palette based on the symbol palette in Stata used in scheme s2mono. This palette supports up to 10 values.

Usage

stata_shape_pal()

Details

This palette uses unicode symbols for the shapes. This means that it will not work for all graphics devices.
See Also

See `scale_shape_stata()` for examples.

---

**stat_fivenumber**

*Calculate components of a five-number summary*

**Description**

The five number summary of a sample is the minimum, first quartile, median, third quartile, and maximum.

**Usage**

```r
stat_fivenumber(
  mapping = NULL,
  data = NULL,
  geom = "boxplot",
  probs = c(0, 0.25, 0.5, 0.75, 1),
  na.rm = FALSE,
  position = "identity",
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

**Arguments**

- **mapping**: Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
- **data**: The data to be displayed in this layer. There are three options:
  - If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`.
  - A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created.
  - A function will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A function can be created from a formula (e.g. `~ head(.x, 10)`).
- **geom**: The geometric object to use to display the data, either as a ggproto Geom subclass or as a string naming the geom stripped of the geom_prefix (e.g. "point" rather than "geom_point")
- **probs**: Quantiles to use for the five number summary.
- **na.rm**: If FALSE (the default), removes missing values with a warning. If TRUE silently removes missing values.
Position adjustment, either as a string naming the adjustment (e.g. "jitter" to use `position_jitter`), or the result of a call to a position adjustment function. Use the latter if you need to change the settings of the adjustment.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn’t inherit behaviour from the default plot specification, e.g. `borders()`.

`...` Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

Value

A data frame with additional columns:

- `width` width of boxplot
- `min` minimum
- `lower` lower hinge, 25% quantile
- `middle` median, 50% quantile
- `upper` upper hinge, 75% quantile
- `max` maximum

See Also

`stat_boxplot()`

---

**tableau_color_pal**  
*Tableau Color Palettes (discrete)*

Description

Color palettes used in Tableau.

Usage

```r
tableau_color_pal(
  palette = "Tableau 10",
  type = c("regular", "ordered-sequential", "ordered-diverging"),
  direction = 1
)
```
Arguments

- **palette**: Palette name. See Details for available palettes.
- **type**: Type of palette. One of "regular", "ordered-diverging", or "ordered-sequential".
- **direction**: If 1, the default, then use the original order of colors. If -1, then reverse the order.

Details

Tableau provides three types of color palettes: "regular" (discrete, qualitative categories), "ordered-sequential", and "ordered-diverging".


References


See Also

Other colour tableau: `scale_colour_gradient2_tableau()`, `scale_colour_gradient_tableau()`, `scale_colour_tableau()`, `tableau_gradient_pal()`
**Examples**

```r
library("scales")

palettes <- ggthemes_data[["tableau"]][["color-palettes"]][["regular"]][
    for (palname in names(palettes)) {
    pal <- tableau_color_pal(palname)
    max_n <- attr(pal, "max_n")
    show_col(pal(max_n))
    title(main = palname)
}
```

---

**tableau_gradient_pal**  
Tableau colour gradient palettes (continuous)

---

**Description**

Gradient color palettes using the diverging and sequential continuous color palettes in Tableau. See `tableau_color_pal()` for discrete color palettes.

**Usage**

```r
tableau_gradient_pal(palette = "Blue", type = "ordered-sequential")

tableau_seq_gradient_pal(palette = "Blue", ...)

tableu_div_gradient_pal(palette = "Orange-Blue Diverging", ...)
```

**Arguments**

- **palette**
  - Palette name. One of the following:

- **type**
  - Palette type, either "ordered-sequential" or "ordered-diverging".

- **...**
  - Arguments passed to `tableau_gradient_pal`. 
See Also

Other colour tableau: `scale_colour_gradient2_tableau()`, `scale_colour_gradient_tableau()`, `scale_colour_tableau()`, `tableau_color_pal()`

Examples

```r
library("scales")

x <- seq(0, 1, length = 25)
r <- sqrt(outer(x ^ 2, x ^ 2, "+"))
palettes <-
  ggthemes_data["tableau"][["color-palettes"]][["ordered-sequential"]]
for (palname in names(palettes)) {
  col <- tableau_seq_gradient_pal(palname)(seq(0, 1, length = 12))
  image(r, col = col)
  title(main = palname)
}
```

---

**tableau_shape_pal**  
*Tableau Shape Palettes (discrete)*

Description

Shape palettes used by Tableau.

Usage

```
tableau_shape_pal(palette = c("default", "filled", "proportions"))
```

Arguments

- `palette`  
  Palette name.

Details

Not all shape palettes in Tableau are supported. Additionally, these palettes are not exact, and use the best unicode character for the shape palette.

Since these palettes use unicode characters, their look may depend on the font being used, and not all characters may be available.

Shape palettes in Tableau are used to expose images for use as markers in charts, and thus are sometimes groupings of closely related symbols.

This palette uses unicode symbols. It may not work on all graphics devices.

See Also

Other shape tableau: `scale_shape_tableau()`
## theme_base

### Description

Theme similar to the default settings of the ‘base’ R graphics.

### Usage

```r
theme_base(base_size = 16, base_family = "")
```

### Arguments

- `base_size`: base font size, given in pts.
- `base_family`: base font family

### See Also

Other themes: `theme_clean()`, `theme_foundation()`, `theme_igray()`, `theme_par()`, `theme_solid()`

### Examples

```r
library("ggplot2")

p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg, colour = factor(gear))) + facet_wrap(~am)
p + theme_base()

# Change values of par
par(fg = "blue", bg = "gray", col.lab = "red", font.lab = 3)
p + theme_base()
```
theme_calc  Theme Calc

Description

Theme similar to the default settings of LibreOffice Calc charts.

Usage

theme_calc(base_size = 10, base_family = "sans")

Arguments

base_size  base font size, given in pts.
base_family  base font family

Examples

## Not run:
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am) + theme_calc()
p + scale_color_calc()

q <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, shape = factor(gear))) +
  facet_wrap(~am) +
  theme_calc()
q + scale_shape_calc()

## End(Not run)

theme_clean  Clean ggplot theme

Description

Clean ggplot theme with no panel background, black axis lines and grey fill colour for chart elements.

Usage

theme_clean(base_size = 12, base_family = "sans")
theme_economist

Arguments

base_size Base font size.
base_family Base font family.

Author(s)

Konrad Zdeb <name.surname@me.com>

See Also

Other themes: theme_base(), theme.foundation(), theme_igray(), theme_par(), theme_solid()

Examples

library("ggplot2")
p <- ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(gear))) + geom_point() + facet_wrap(~am) + geom_smooth(method = "lm", se = FALSE) + theme_clean()
p

theme_economist ggplot color theme based on the Economist

Description

A theme that approximates the style of The Economist.

Usage

theme_economist(
  base_size = 10,
  base_family = "sans",
  horizontal = TRUE,
  dkpanel = FALSE
)

theme_economist_white(
  base_size = 11,
  base_family = "sans",
  gray_bg = TRUE,
  horizontal = TRUE
)
Arguments

base_size  base font size, given in pts.
base_family base font family
horizontal logical Horizontal axis lines?
dkpanel logical Darker background for panel region?
gray_bg logical If TRUE, use gray background, else use white background.

Details

theme_economist implements the standard bluish-gray background theme in the print The Economist and economist.com.

theme_economist_white implements a variant with a while panel and light gray (or white) background often used by The Economist blog Graphic Detail.

Use scale_color_economist() with this theme. The x axis should be displayed on the right hand side.

The Economist uses "ITC Officina Sans" as its font for graphs. If you have access to this font, you can use it with the extrafont package. "Verdana" is a good substitute.

Value

An object of class theme().

References

- The Economist

Examples

library("ggplot2")

p <- ggplot(mtcars) +
    geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
    facet_wrap(~am) +
    # Economist puts x-axis labels on the right-hand side
    scale_y_continuous(position = "right")

## Standard
p + theme_economist() +
    scale_colour_economist()

# Change axis lines to vertical
p + theme_economist(horizontal = FALSE) +
    scale_colour_economist() +
    coord_flip()

## White panel/light gray background
p + theme_economist_white() +
scale_colour_economist()

## All white variant
p + theme_economist_white(gray_bg = FALSE) +
    scale_colour_economist()

## Not run:
## The Economist uses ITC Officina Sans
library("extrafont")
p + theme_economist(base_family="ITC Officina Sans") +
    scale_colour_economist()

## Verdana is a widely available substitute
p + theme_economist(base_family="Verdana") +
    scale_colour_economist()

## End(Not run)

---

theme_excel  

**ggplot theme based on old Excel plots**

**Description**

Theme to replicate the ugly monstrosity that was the old gray-background Excel chart. Please never use this. This theme should be combined with the `scale_colour_excel()` color scale.

**Usage**

```r
theme_excel(base_size = 12, base_family = "", horizontal = TRUE)
```

**Arguments**

- `base_size`: base font size, given in pts.
- `base_family`: base font family
- `horizontal`: logical. Horizontal axis lines?

**Value**

An object of class `theme()`.

**See Also**

Other themes excel: `theme_excel_new()`
Examples

```r
library("ggplot2")

# Line and scatter plot colors
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)

p + theme_excel() + scale_colour_excel()

# Bar plot (area/fill) colors

ggplot(mpg, aes(x = class, fill = drv)) +
  geom_bar() +
  scale_fill_excel() +
  theme_excel()
```

theme_excel_new

ggplot theme similar to current Excel plot defaults

Description

Theme for ggplot2 that is similar to the default style of charts in current versions of Microsoft Excel.

Usage

```r
theme_excel_new(base_size = 9, base_family = "sans")
```

Arguments

- `base_size`: base font size, given in pts.
- `base_family`: base font family

Value

An object of class `theme()`.

See Also

Other themes excel: `theme_excel()`

Examples

```r
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)

p + theme_excel_new() + scale_colour_excel_new()
```
**theme_few**

*Theme based on Few’s “Practical Rules for Using Color in Charts”*

### Description

Theme based on the rules and examples from Stephen Few’s *Show Me the Numbers* and "Practical Rules for Using Color in Charts".

### Usage

```r
theme_few(base_size = 12, base_family = "")
```

### Arguments

- `base_size`: base font size, given in pts.
- `base_family`: base font family

### References


### Examples

```r
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)
p + theme_few() + scale_colour_few()
p + theme_few() + scale_colour_few("Light")
p + theme_few() + scale_colour_few("Dark")

ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, shape = factor(gear))) +
  theme_few() +
  scale_shape_few()
```
theme_fivethirtyeight  Theme inspired by FiveThirtyEight plots

Description

Theme inspired by the plots that were produced by the site FiveThirtyEight (fivethirtyeight.com).

Usage

```r
theme_fivethirtyeight(base_size = 12, base_family = "sans")
```

Arguments

- `base_size`: base font size, given in pts.
- `base_family`: base font family

Examples

```r
library("ggplot2")
p <- ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(gear))) +
geom_point() +
facet_wrap(~am) +
geom_smooth(method = "lm", se = FALSE) +
scale_color_fivethirtyeight() +
theme_fivethirtyeight()
p
```

defeated

theme_foundation  Foundation Theme

Description

This theme is designed to be a foundation from which to build new themes, and not meant to be used directly. theme_foundation() is a complete theme with only minimal number of elements defined. It is easier to create new themes by extending this one rather than theme_gray() or theme_bw(), because those themes define elements deep in the hierarchy.

Usage

```r
theme_foundation(base_size = 12, base_family = "")
```

Arguments

- `base_size`: base font size, given in pts.
- `base_family`: base font family
**theme_gdocs**

**Details**

This theme takes `theme_gray()` and sets all colour and fill values to NULL, except for the top-level elements (line, rect, and title), which have colour = "black", and fill = "white". This leaves the spacing and non-colour defaults of the default `ggplot2` themes in place.

**See Also**

Other themes: `theme_base()`, `theme_clean()`, `theme_igray()`, `theme_par()`, `theme_solid()`

---

**theme_gdocs**  
*Theme with Google Docs Chart defaults*

**Description**

Theme similar to the default look of charts in Google Docs.

**Usage**

```r
theme_gdocs(base_size = 12, base_family = "sans")
```

**Arguments**

- `base_size`  
  base font size, given in pts.
- `base_family`  
  base font family

**Examples**

```r
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am)

p + theme_gdocs() + scale_color_gdocs()
```

---

**theme_hc**  
*Highcharts Theme*

**Description**

Theme based on Highcharts plots.
Usage

```r
theme_hc(
  base_size = 12,
  base_family = "sans",
  style = c("default", "darkunica"),
  bgcolor = NULL
)
```

Arguments

- **base_size**: base font size, given in pts.
- **base_family**: base font family
- **style**: The Highcharts theme to use 'default', 'darkunica'.
- **bgcolor**: Deprecated

References

https://www.highcharts.com/demo/highcharts/line-chart

Examples

```r
library("ggplot2")

p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg, 
  colour = factor(gear))) + facet_wrap(~am)
p + theme_hc() + scale_colour_hc()
p + theme_hc(bgcolor = "darkunica") + 
  scale_colour_hc("darkunica")

dtemp <- data.frame(months = factor(rep(substr(month.name, 1, 3), 4), 
  levels = substr(month.name, 1, 3)), 
  city = rep(c("Tokyo", "New York", "Berlin", "London"), 
  each = 12), 
  temp = c(7.0, 6.9, 9.5, 14.5, 18.2, 21.5, 
  25.2, 26.5, 23.3, 18.3, 13.9, 9.6, 
  -0.2, 0.8, 5.7, 11.3, 17.0, 22.0, 
  24.8, 24.1, 20.1, 14.1, 8.6, 2.5, 
  -0.9, 0.6, 3.5, 8.4, 13.5, 17.0, 
  18.6, 17.9, 14.3, 9.0, 3.9, 1.0, 
  3.9, 4.2, 5.7, 8.5, 11.9, 15.2, 
  17.0, 16.6, 14.2, 10.3, 6.6, 4.8))

ggplot(dtemp, aes(x = months, y = temp, group = city, color = city)) + 
  geom_line() + 
  geom_point(size = 1.1) + 
  ggtitle("Monthly Average Temperature") + 
  theme_hc() + 
  scale_colour_hc()

 ggplot(dtemp, aes(x = months, y = temp, group = city, color = city)) + 
```
theme_igray

```r
ggplot2

geom_line() +
geom_point(size = 1.1) +
ggtitle("Monthly Average Temperature") +
theme_hc(bgcolor = "darkunica") +
scale_fill_hc("darkunica")
```

---

**theme_igray**  
**Inverse gray theme**

**Description**

Theme with white panel and gray background.

**Usage**

```r
theme_igray(base_size = 12, base_family = "")
```

**Arguments**

- `base_size`: base font size, given in pts.
- `base_family`: base font family

**Details**

This theme inverts the colors in the `theme_gray()`, a white panel and a light gray area around it. This keeps a white background for the color scales like `theme_bw()`. But by using a gray background, the plot is closer to the typographical color of the document, which is the motivation for using a gray panel in `theme_gray()`. This is similar to the style of plots in Stata and Tableau.

**See Also**

`theme_gray()`, `theme_bw()`

Other themes: `theme_base()`, `theme_clean()`, `theme.foundation()`, `theme_par()`, `theme.solid()`

**Examples**

```r
library("ggplot2")

p <- ggplot(mtcars) +
    geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
    facet_wrap(~am)

p + theme_igray()
```
theme_map

Clean theme for maps

Description

A clean theme that is good for displaying maps from `geom_map()`.

Usage

```r
theme_map(base_size = 9, base_family = "")
```

Arguments

- `base_size` base font size, given in pts.
- `base_family` base font family

Examples

```r
library("maps")
library("ggplot2")

us <- fortify(map_data("state"), region = "region")
gg <- ggplot() +
  geom_map(data = us, map = us, 
    aes(x = long, y = lat, map_id = region, group = group),
    fill = "white", color = "black", size = 0.25) +
  coord_map("albers", lat0 = 39, lat1 = 45) +
  theme_map()
gg
```

theme_pander

A `ggplot` theme originated from the `pander` package

Description

The `pander` ships with a default theme when the 'unify plots' option is enabled via `panderOptions`, which is now also available outside of `pander` internals, like `evals`, `evalmsgs` or `Pandoc.brew`.

Usage

```r
theme_pander(
  base_size = 12,
  base_family = "sans",
  nomargin = TRUE,
  ff = NULL,
  fc = "black",
```
theme_pander

fs = NULL,
gM = TRUE,
gm = TRUE,
gc = "grey",
gl = "dashed",
boxes = FALSE,
bc = "white",
pc = "transparent",
lp = "right",
axis = 1
)

Arguments

base_size base font size, given in pts.
base_family base font family
nomargin suppress the white space around the plot (boolean)
ff font family, like sans. Deprecated: use base_family instead.
fc font color (name or hexa code)
fs font size (integer). Deprecated: use base_size instead.
gM major grid (boolean)
gm minor grid (boolean)
gc grid color (name or hexa code)
gl grid line type (lty)
boxes to render a border around the plot or not
bc background color (name or hexa code)
pc panel background color (name or hexa code)
lp legend position
axis axis angle as defined in par(les)

Examples

require("ggplot2")
if (require("pander")) {
  p <- ggplot(mtcars, aes(x = mpg, y = wt)) +
    geom_point()
  p + theme_pander()

  panderOptions("graph.grid.color", "red")
  p + theme_pander()

  p <- ggplot(mtcars, aes(wt, mpg, colour = factor(cyl))) +
    geom_point()
  p + theme_pander() + scale_color_pander()
```r
ggplot(mpg, aes(x = class, fill = drv)) + geom_bar() + scale_fill_pander() + theme_pander()
```

**theme_par**

Theme which uses the current `base` graphics parameter values from `par()`. Not all `par()` parameters, are supported, and not all are relevant to `ggplot2` themes.

**Description**


**Usage**

```r
theme_par(base_size = par()$ps, base_family = par()$family)
```

**Arguments**

- `base_size` base font size, given in pts.
- `base_family` base font family

**Details**

This theme does not translate the base graphics perfectly, so the graphs produced by it will not be identical to those produced by base graphics, most notably in the spacing of the margins.

**See Also**

Other themes: `theme_base()`, `theme_clean()`, `theme_foundation()`, `theme_igray()`, `theme_solid()`

**Examples**

```r
library("ggplot2")

p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg, colour = factor(gear))) + facet_wrap(~am)

p + theme_par()

# theme changes with respect to values of par
par(font = 2, col.lab = "red", fg = "white", bg = "black")
p + theme_par()```
theme_solarized

**Description**


**Usage**

```r
theme_solarized(base_size = 12, base_family = "", light = TRUE)
theme_solarized_2(base_size = 12, base_family = "", light = TRUE)
```

**Arguments**

- `base_size`: base font size, given in pts.
- `base_family`: base font family
- `light`: logical. Light or dark theme?

**Details**

Plots made with this theme integrate seamlessly with the Solarized Beamer color theme. [https://github.com/jrnold/beamercolorthem.esolarized](https://github.com/jrnold/beamercolorthem.esolarized). There are two variations: `theme_solarized` is similar to `theme_bw()`, while `theme_solarized_2()` is similar to `theme_gray()`.

**Examples**

```r
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear)))

# Light version with different main accent colors
for (accent in names(ggthemes::ggthemes_data[["solarized"]][["accents"]])) {
  print(p + theme_solarized() +
        scale_colour_solarized(accent))
}

# Dark version
p + theme_solarized(light = FALSE) +
  scale_colour_solarized("blue")

# Alternative theme
p + theme_solarized_2(light = FALSE) +
  scale_colour_solarized("blue")
```
theme_solid  

Theme with nothing other than a background color

Description

Theme that removes all non-geom elements (lines, text, etc). This theme is when only the geometric objects are desired.

Usage

theme_solid(base_size = 12, base_family = "", fill = NA)

Arguments

- base_size: Base font size.
- base_family: Ignored, kept for consistency with theme().
- fill: Background color of the plot.

See Also

Other themes: theme_base(), theme_clean(), theme_foundation(), theme_igray(), theme_par()

Examples

library("ggplot2")

ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_solid(fill = "white")

ggplot(mtcars, aes(wt, mpg)) +
  geom_point(color = "white") +
  theme_solid(fill = "black")

theme_stata  

Themes based on Stata graph schemes

Description

Themes based on Stata graph schemes

Usage

theme_stata(base_size = 11, base_family = "sans", scheme = "s2color")
**theme_stata**

**Arguments**

- **base_size** base font size, given in pts.
- **base_family** base font family
- **scheme** One of "s2color", "s2mono", "s1color", "s1rcolor", or "s1mono", "s2manual", "s1manual", or "sj"

**Details**

These themes approximate Stata schemes using the features **ggplot2**. The graphical models of Stata and ggplot2 differ in various ways that make an exact replication impossible (or more difficult than it is worth). Some features in Stata schemes not in ggplot2: defaults for specific graph types, different levels of titles, captions and notes. These themes also adopt some of the ggplot2 defaults, and more effort was made to match the colors and sizes of major elements than in matching the margins.

**References**


**Examples**

```r
library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~ am) +
  labs(title = "Graphs by car type",
       x = "Weight (lbs.)", y = "MPG")

# s2color
p + theme_stata() +
  scale_colour_stata("s2color")
# s2mono
p + theme_stata(scheme = "s2mono") +
  scale_colour_stata("mono")
# s1color
p + theme_stata(scheme = "s2color") +
  scale_colour_stata("s1color")
# s1rcolor
p + theme_stata(scheme = "s1rcolor") +
  scale_colour_stata("s1rcolor")
# s1mono
p + theme_stata(scheme = "s1mono") +
  scale_colour_stata("mono")
```
theme_tufte

Tufte Maximal Data, Minimal Ink Theme

Description
Theme based on Chapter 6 'Data-Ink Maximization and Graphical Design' of Edward Tufte *The Visual Display of Quantitative Information*. No border, no axis lines, no grids. This theme works best in combination with `geom_rug()` or `geom_rangeframe()`.

Usage

```r
theme_tufte(base_size = 11, base_family = "serif", ticks = TRUE)
```

Arguments

- `base_size` base font size, given in pts.
- `base_family` base font family
- `ticks` logical Show axis ticks?

Note
The default font family is set to 'serif' as he uses serif fonts for labels in 'The Visual Display of Quantitative Information'. The serif font used by Tufte in his books is a variant of Bembo, while the sans serif font is Gill Sans. If these fonts are installed on your system, then you can use them with the package `extrafont`.

References

Examples

```r
library("ggplot2")

p <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  scale_x_continuous(breaks = extended_range_breaks()(mtcars$wt)) +
  scale_y_continuous(breaks = extended_range_breaks()(mtcars$mpg)) +
  ggtitle("Cars")

p + geom_rangeframe() +
  coord_cartesian(clip="off") +
  theme_tufte()

p + geom_rug() +
  theme_tufte(ticks = FALSE)
```
theme_wsj

Wall Street Journal theme

Description

Theme based on the plots in The Wall Street Journal.

Usage

theme_wsj(
  base_size = 12,
  color = "brown",
  base_family = "sans",
  title_family = "mono"
)

Arguments

base_size  base font size, given in pts.
color      The background color of plot. One of 'brown', 'gray', 'green', 'blue'.
base_family base font family
title_family Plot title font family.

Details

This theme should be used with scale_color_wsj().

References

https://twitter.com/WSJGraphics
https://pinterest.com/wsjgraphics/wsj-graphics/

Examples

library("ggplot2")

p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour = factor(gear))) +
  facet_wrap(~am) +
  ggtitle("Diamond Prices")
p + scale_colour_wsj("colors6", "") + theme_wsj()
# Use a gray background instead
p + scale_colour_wsj("colors6", "") + theme_wsj(color = "gray")
tremmel_shape_pal  
*Shape palette from Tremmel (1995) (discrete)*

### Description

Based on experiments Tremmel (1995) suggests the following shape palettes:

If two symbols, then use a solid circle and plus sign.

If three symbols, then use a solid circle, empty circle, and an empty triangle. However, that set of symbols does not satisfy the requirement that each symbol should differ from the other symbols in the same feature dimension. A set of three symbols that satisfies this is a circle (curvature), plus sign (number of terminators), triangle (line orientation).

This palette supports up to three values. If more than three groups of data, then separate the groups into different plots.

Not work for all graphics devices.

### Usage

`tremmel_shape_pal(overlap = FALSE, alt = FALSE, n3alt = deprecated())`

### Arguments

- **overlap**: use an empty circle instead of a solid circle when \(n = 2\).
- **alt**: If TRUE, then when \(n = 3\), use a solid circle, plus sign and empty triangle. Otherwise use a solid circle, empty circle, and empty triangle.
- **n3alt**: 'r lifecycle::badge("deprecated")' 'n3alt' is no longer supported; use 'alt' instead.

### References


### See Also

Other shapes: `circlefill_shape_pal()`, `cleveland_shape_pal()`, `scale_shape_circlefill()`, `scale_shape_cleveland()`, `scale_shape_tremmel()`
Description

The Wall Street Journal uses many different color palettes in its plots. This collects a few of them, but is by no means exhaustive. Collections of these plots were found in the WSJ Graphics Pinterest collection.

Usage

wsj_pal(palette = "colors6")

Arguments

palette character The color palette to use: "rgby", "red_green", "black_green", "dem_rep", "colors6"

Palettes

The following palettes are defined,

rgby Red/Green/Blue/Yellow theme. Up to four values.
red_green Green/red two-color scale for good/bad.
green_black Black-green 4-color scale for 'Very negative', 'Somewhat negative', 'somewhat positive', 'very positive'.
dem_rep Democrat/Republican/Undecided blue/red/gray scale.
colors6 Red, blue, gold, green, orange, and black palette.

See Also

Other colour wsj: scale_colour_wsj()
Index

* colour calc
calc_pal, 6
scale_fill_calc, 46
* colour economist
economist_pal, 12
scale_colour_economist, 28
* colour excel
excel_new_pal, 13
excel_pal, 14
scale_colour_excel_new, 29
scale_fill_excel, 47
* colour few
few_pal, 16
scale_colour_few, 31
* colour fivethirtyeight
fivethirtyeight_pal, 17
scale_colour_fivethirtyeight, 33
* colour gdocs
gdocs_pal, 18
scale_fill_gdocs, 49
* colour hc
hc_pal, 24
scale_colour_hc, 38
* colour pander
palette_pander, 25
scale_color_pander, 26
* colour ptol
ptol_pal, 25
scale_colour_ptol, 40
* colour scales
scale_fill_solarized, 51
* colour stata
scale_colour_stata, 41
* colour tableau
scale_colour_gradient2_tableau, 34
scale_colour_gradient_tableau, 36
scale_colour_tableau, 43
tableau_color_pal, 69
tableau_gradient_pal, 71
* colour wsj
scale_colour_wsj, 44
wsj_pal, 93
* colour
colorblind_pal, 11
* datasets
canva_palettes, 8
geom_rangeframe, 18
geom_tufteboxplot, 20
ggthemes_data, 24
stat_fivenumber, 68
* geom tuftef
geom_rangeframe, 18
geom_tufteboxplot, 20
* linetype stata
scale_linetype_stata, 52
stata_linetype_pal, 66
* shape stata
scale_shape_stata, 59
* shape tableau
scale_shape_tableau, 60
tableau_shape_pal, 72
* shapes calc
calc_shape_pal, 6
scale_shape_calc, 54
* shapes stata
stata_shape_pal, 67
* shapes
circlefill_shape_pal, 9
cleveland_shape_pal, 9
scale_shape_circlefill, 55
scale_shape_cleveland, 57
scale_shape_tremmel, 62
tremmel_shape_pal, 92
* solarized colour
scale_fill_solarized, 51
solarized_pal, 66
* stata colour
stata_pal, 67
INDEX

* themes calc
  theme_calc, 74
* themes economist
  theme_economist, 75
* themes excel
  theme_excel, 77
  theme_excel_new, 78
* themes few
  theme_few, 79
* themes fivethirtyeight
  theme_fivethirtyeight, 80
* themes gdocs
  theme_gdocs, 81
* themes hc
  theme_hc, 81
* themes solarized
  theme_solarized, 87
* themes stata
  theme_stata, 88
* themes tufte
  theme_tufte, 90
* themes wsj
  theme_wsj, 91
* themes
  theme_base, 73
  theme_clean, 74
  theme_foundation, 80
  theme_igray, 83
  theme_par, 86
  theme_solid, 88

aes(), 19, 21, 68

bank_slopes, 4
banking, 5
borders(), 19, 22, 69

calc_pal, 6, 47
calc_shape_pal, 6, 54, 55
canva_pal, 7
canva_palettes, 7, 8, 27, 28
circlefill_shape_pal, 9, 10, 56–58, 63, 92
cleveland_shape_pal, 9, 9, 57, 58, 63, 92
colorblind_pal, 11
dichromat_pal, 12
discrete_scale, 11, 26, 28, 30, 32, 33, 38, 40, 42, 45, 46, 48, 49, 51, 53, 54, 56–60, 62
discrete_scale(), 44

economist_pal, 12, 29
eexcel_new_pal, 13, 14, 31, 49
eexcel_pal, 13, 14, 31, 49
expansion(), 12, 27, 29, 31, 32, 34, 39, 41, 43, 45, 47, 48, 50, 52
extended_range_breaks
  (extended_range_breaks_), 14
extended_range_breaks_, 14
few_pal, 16, 31, 33
few_shape_pal, 17
fivethirtyeight_pal, 17, 34
format, 65
fortify(), 19, 21, 68
gdocs_pal, 18, 50
gem_boxplot, 22
gem_map, 84
gem_rangeframe, 18, 22, 90
gem_rug, 90
gem_tufteboxplot, 20, 20
GeomRangeFrame (geom_rangeframe), 18
GeomTufteboxplot (geom_tufteboxplot), 20
ggplot(), 19, 21, 68
ggthemes, 23
ggthemes_package (ggthemes), 23
ggthemes_data, 24
guides(), 12, 27, 29, 31, 32, 34, 39, 41, 43, 45, 47, 48, 50, 52

hc_pal, 24, 38, 39

lambda, 11, 12, 26–34, 39–42, 45–63
layer(), 19, 22, 69

palette_pander, 25, 27
par, 64, 86
ptol_pal, 25, 40, 41

scale_color_calc (scale_fill_calc), 46
scale_color_canva (scale_colour_canva), 27
scale_color_continuous_tableau
  (scale_colour_gradient_tableau), 36
scale_color_economist, 76
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>scale_color_economist</td>
<td>28</td>
</tr>
<tr>
<td>scale_color_excel (scale_fill_excel)</td>
<td>47</td>
</tr>
<tr>
<td>scale_color_excel_new (scale_colour_excel_new)</td>
<td>29</td>
</tr>
<tr>
<td>scale_color_few (scale_colour_few)</td>
<td>31</td>
</tr>
<tr>
<td>scale_color_fivethirtyeight</td>
<td>33</td>
</tr>
<tr>
<td>scale_color_gdocs (scale_fill_gdocs)</td>
<td>49</td>
</tr>
<tr>
<td>scale_color_gradient2_tableau (scale_colour_gradient2_tableau)</td>
<td>34</td>
</tr>
<tr>
<td>scale_color_gradient_tableau (scale_colour_gradient_tableau)</td>
<td>36</td>
</tr>
<tr>
<td>scale_color_hc (scale_colour_hc)</td>
<td>38</td>
</tr>
<tr>
<td>scale_color_pander (scale_color_pander)</td>
<td>26</td>
</tr>
<tr>
<td>scale_color_ptol (scale_colour_ptol)</td>
<td>40</td>
</tr>
<tr>
<td>scale_color_solarized (scale_fill_solarized)</td>
<td>51</td>
</tr>
<tr>
<td>scale_color_stat (scale_colour_stat)</td>
<td>41</td>
</tr>
<tr>
<td>scale_color_tableau (scale_colour_tableau)</td>
<td>12</td>
</tr>
<tr>
<td>scale_color_tableau (scale_colour_tableau)</td>
<td>43</td>
</tr>
<tr>
<td>scale_color_wsj (scale_colour_wsj)</td>
<td>91</td>
</tr>
<tr>
<td>scale_color_wsj (scale_colour_wsj)</td>
<td>44</td>
</tr>
<tr>
<td>scale_color_calc (scale_fill_calc)</td>
<td>46</td>
</tr>
<tr>
<td>scale_color_canva</td>
<td>27</td>
</tr>
<tr>
<td>scale_colour_colorblind (colorblind_pal)</td>
<td>11</td>
</tr>
<tr>
<td>scale_colour_economist</td>
<td>28</td>
</tr>
<tr>
<td>scale_colour_excel</td>
<td>47</td>
</tr>
<tr>
<td>scale_colour_excel_new</td>
<td>29</td>
</tr>
<tr>
<td>scale_colour_few</td>
<td>31</td>
</tr>
<tr>
<td>scale_colour_fivethirtyeight</td>
<td>33</td>
</tr>
<tr>
<td>scale_color_gdocs</td>
<td>49</td>
</tr>
<tr>
<td>scale_color_gradient2_tableau</td>
<td>34, 38, 44, 70, 72</td>
</tr>
<tr>
<td>scale_color_gradient2_tableau</td>
<td>34, 38, 44, 70, 72</td>
</tr>
<tr>
<td>scale_color_gradient2_tableau()</td>
<td>36, 43</td>
</tr>
<tr>
<td>scale_color_gradient_tableau</td>
<td>36, 36, 44, 70, 72</td>
</tr>
<tr>
<td>scale_color_gradient_tableau()</td>
<td>34, 43</td>
</tr>
<tr>
<td>scale_color_hc</td>
<td>24, 38</td>
</tr>
<tr>
<td>scale_color_pander</td>
<td>26</td>
</tr>
<tr>
<td>scale_color_ptol</td>
<td>26, 40</td>
</tr>
<tr>
<td>scale_color_solarized</td>
<td>51</td>
</tr>
<tr>
<td>scale_colour_stat</td>
<td>41</td>
</tr>
<tr>
<td>scale_colour_tableau</td>
<td>36, 38, 43, 70, 72</td>
</tr>
<tr>
<td>scale_colour_wsj</td>
<td>44, 93</td>
</tr>
<tr>
<td>scale_fill_cal</td>
<td>6, 46</td>
</tr>
<tr>
<td>scale_fill_canva (scale_colour_canva)</td>
<td>27</td>
</tr>
<tr>
<td>scale_fill_colorblind (colorblind_pal)</td>
<td>11</td>
</tr>
<tr>
<td>scale_fill_continuous_tableau</td>
<td>36</td>
</tr>
<tr>
<td>scale_fill_economist</td>
<td>28</td>
</tr>
<tr>
<td>scale_fill_excel</td>
<td>13, 14, 31, 47</td>
</tr>
<tr>
<td>scale_fill_excel_new</td>
<td>29</td>
</tr>
<tr>
<td>scale_fill_few</td>
<td>31</td>
</tr>
<tr>
<td>scale_fill_fivethirtyeight</td>
<td>33</td>
</tr>
<tr>
<td>scale_fill_gdocs</td>
<td>18, 49</td>
</tr>
<tr>
<td>scale_fill_gradient2_tableau</td>
<td>34</td>
</tr>
<tr>
<td>scale_fill_gradient_tableau</td>
<td>36</td>
</tr>
<tr>
<td>scale_fill_hc</td>
<td>38</td>
</tr>
<tr>
<td>scale_fill_pander</td>
<td>26</td>
</tr>
<tr>
<td>scale_fill_ptol (scale_color_pander)</td>
<td>40</td>
</tr>
<tr>
<td>scale_fill_solarized</td>
<td>51, 66</td>
</tr>
<tr>
<td>scale_fill_stata (scale_colour_stata)</td>
<td>41</td>
</tr>
<tr>
<td>scale_fill_tableau</td>
<td>43</td>
</tr>
<tr>
<td>scale_fill_wsj (scale_colour_wsj)</td>
<td>44</td>
</tr>
<tr>
<td>scale_linetype_stata</td>
<td>52, 66</td>
</tr>
<tr>
<td>scale_shape_calc</td>
<td>54</td>
</tr>
<tr>
<td>scale_shape_circlefill</td>
<td>9, 10, 55, 58, 63, 92</td>
</tr>
</tbody>
</table>
INDEX

scale_shape_cleveland, 9, 10, 57, 63, 92
scale_shape_few, 58, 58
scale_shape_stata, 59, 68
scale_shape_tableau, 60, 72
scale_shape_tremmel, 9, 10, 57, 58, 62, 92
scales::hue_pal(), 11, 26, 28, 30, 32, 33, 38, 40, 42, 45, 46, 48, 49, 51, 53, 54, 56, 57, 59, 60, 62
show_col, 64
show_linetypes, 63, 64
show_shapes, 64
smart_digits, 65
smart_digits_format(smart_digits), 65
solarized_pal, 51, 52, 66
stat_boxplot, 69
stat_fivenumber, 68
stata_linetype_pal, 52, 53, 66
stata_pal, 41, 67
stata_shape_pal, 59, 67
StatFivenumber(stat_fivenumber), 68
tableau_color_pal, 36, 38, 44, 69, 72
tableau_color_pal(), 44, 71
tableau_div_gradient_pal
  (tableau_gradient_pal), 71
tableau_gradient_pal, 36, 38, 44, 70, 71
tableau_seq_gradient_pal
  (tableau_gradient_pal), 71
tableau_shape_pal, 60, 61, 72
theme, 76–78, 88
theme_base, 73, 75, 81, 83, 86, 88
theme_bw, 80, 83, 87
theme_calc, 47, 55, 74
theme_clean, 73, 74, 81, 83, 86, 88
theme_economist, 29, 75
theme_economist_white
  (theme_economist), 75
theme_excel, 77, 78
theme_excel_new, 77, 78
theme_few, 79
theme_fivethirtyeight, 34, 80
theme_foundation, 73, 75, 80, 83, 86, 88
theme_gdocs, 50, 81
theme_gray, 80, 81, 83, 87
theme_hc, 38, 81
theme_igray, 73, 75, 81, 83, 86, 88
theme_map, 84
theme_pander, 27, 84
theme_par, 73, 75, 81, 83, 86, 88
theme_solarized, 87
theme_solarized_2(theme_solarized), 87
themeSolid, 73, 75, 81, 83, 86, 88
theme_stata, 88
theme_tufte, 90
theme_wsj, 44, 91
tremmel_shape_pal, 9, 10, 57, 58, 63, 92
wsj_pal, 44, 46, 93