Package ‘ggthemes’

February 19, 2017

Version 3.4.0

Title Extra Themes, Scales and Geoms for 'ggplot2'

Depends R (>= 3.0.0),

Imports assertthat, colorspace, ggplot2 (>= 2.2.0), graphics, grid,
       methods, scales

Suggests extrafont, knitr, lintr, maps, mapproj, pander, plyr,
       reshape2, rmarkdown, testthat, tidyverse

VignetteBuilder knitr

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 http://github.com/jrnold/ggthemes

BugReports http://github.com/jrnold/ggthemes

Collate 'banking.R' 'base.R' 'calc.R' 'canva.R' 'colorblind.R'
    'economist.R' 'excel.R' 'few.R' 'ggthemes-data.R'
    'ggthemes-package.R' 'fivethirtyeight.R' 'gdocs.R'
    'geom-rangeframe.R' 'geom-tufteboxplot.R' 'hc.R' 'igray.R'
    'pander.R' 'ptol.R' 'scale-tufte.R' 'shapes.R' 'show.R'
    'solarized.R' 'stat-fivenumber.R' 'stata.R' 'tableau.R'
    'theme-foundation.R' 'theme-map.R' 'theme-solid.R' 'tufte.R'
    'utils.R' 'wsj.R'

RoxygenNote 6.0.1

LazyData true

NeedsCompilation no

Author Jeffrey B. Arnold [aut, cre],
    Gergely Daroczi [ctb],
    Bo Werth [ctb],
    Brian Weitzner [ctb],

1
Joshua Kunst [ctb],
Baptise Auguie [ctb],
Bob Rudis [ctb],
Hadley Wickham [ctb, cph] (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 2017-02-19 10:53:38

R topics documented:

bank_slopes ........................................ 4
calc_pal ............................................. 6
calc_shape_pal .................................... 6
canva_pal .......................................... 7
canva_palettes ................................... 7
circlefill_shape_pal ............................... 8
cleveland_shape_pal ............................... 9
colorblind_pal ..................................... 10
economist_pal ..................................... 11
excel_pal .......................................... 12
extended_range_breaks ............................ 12
few_pal ............................................ 14
fivethirtyeight_pal ................................. 14
gdocs_pal .......................................... 15
geom_rangeframe ................................. 15
geom_tufteboxplot ................................ 17
ggthemes ........................................... 19
ggthemes_data ..................................... 19
hc_pal ............................................. 20
palette_pander .................................. 20
ptol_pal ........................................... 21
scale_color_pander ............................... 22
scale_colour_canva ............................... 22
scale_colour_economist ......................... 23
scale_colour_few ................................ 23
scale_colour_fivethirtyeight ................... 24
scale_colour_gradient2_tableau ............... 25
scale_colour_gradient_tableau ................ 26
scale_colour_hc ................................ 27
scale_colour_ptol ................................. 27
scale_colour_stata ............................... 28
scale_colour_tableau ............................. 29
scale_colour_wsj ................................ 30
scale_fill_calc .................................. 30
scale_fill_excel .................................. 31
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>scale_fill_gdocs</td>
<td>31</td>
</tr>
<tr>
<td>scale_fill_solarized</td>
<td>32</td>
</tr>
<tr>
<td>scale_linetype_stata</td>
<td>33</td>
</tr>
<tr>
<td>scale_shape_calc</td>
<td>33</td>
</tr>
<tr>
<td>scale_shape_circlefill</td>
<td>34</td>
</tr>
<tr>
<td>scale_shape_cleveland</td>
<td>34</td>
</tr>
<tr>
<td>scale_shape_stata</td>
<td>35</td>
</tr>
<tr>
<td>scale_shape_tableau</td>
<td>35</td>
</tr>
<tr>
<td>scale_shape_tremmel</td>
<td>36</td>
</tr>
<tr>
<td>show_linetypes</td>
<td>37</td>
</tr>
<tr>
<td>show_shapes</td>
<td>38</td>
</tr>
<tr>
<td>smart_digits</td>
<td>38</td>
</tr>
<tr>
<td>solarized_pal</td>
<td>39</td>
</tr>
<tr>
<td>stata_linetype_pal</td>
<td>40</td>
</tr>
<tr>
<td>stata_pal</td>
<td>40</td>
</tr>
<tr>
<td>stata_shape_pal</td>
<td>41</td>
</tr>
<tr>
<td>stat_fivenumber</td>
<td>41</td>
</tr>
<tr>
<td>tableau_color_pal</td>
<td>42</td>
</tr>
<tr>
<td>tableau_div_gradient_pal</td>
<td>43</td>
</tr>
<tr>
<td>tableau_seq_gradient_pal</td>
<td>44</td>
</tr>
<tr>
<td>tableau_shape_pal</td>
<td>45</td>
</tr>
<tr>
<td>theme_base</td>
<td>45</td>
</tr>
<tr>
<td>theme_calc</td>
<td>46</td>
</tr>
<tr>
<td>theme_economist</td>
<td>46</td>
</tr>
<tr>
<td>theme_excel</td>
<td>48</td>
</tr>
<tr>
<td>theme_few</td>
<td>49</td>
</tr>
<tr>
<td>theme_fivethirtyeight</td>
<td>50</td>
</tr>
<tr>
<td>theme_foundation</td>
<td>50</td>
</tr>
<tr>
<td>theme_gdocs</td>
<td>51</td>
</tr>
<tr>
<td>theme_hc</td>
<td>51</td>
</tr>
<tr>
<td>theme_igray</td>
<td>52</td>
</tr>
<tr>
<td>theme_map</td>
<td>53</td>
</tr>
<tr>
<td>theme_pander</td>
<td>54</td>
</tr>
<tr>
<td>theme_par</td>
<td>55</td>
</tr>
<tr>
<td>theme_solarized</td>
<td>56</td>
</tr>
<tr>
<td>theme固体</td>
<td>56</td>
</tr>
<tr>
<td>theme_stata</td>
<td>57</td>
</tr>
<tr>
<td>theme_tufte</td>
<td>58</td>
</tr>
<tr>
<td>theme_wsj</td>
<td>59</td>
</tr>
<tr>
<td>tremmel_shape_pal</td>
<td>60</td>
</tr>
<tr>
<td>wjs_pal</td>
<td>61</td>
</tr>
</tbody>
</table>

Index 62
**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**

```r
bank_slopes(x, y, cull = FALSE, weight = NULL, method = c("ms", "as", "ao", "gor", "lor"), ...)
```

**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,

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

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.

**Usage**

```r
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**

```r
calc_shape_pal()
```

**See Also**

Other shapes calc: `scale_shape_calc`

**Examples**

```r
library("ggplot2")
show_shapes(calc_shape_pal()(15))
```
**canva_pal**  
*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("scales")
show_col(canva_pal("Fresh and bright")(4))
show_col(canva_pal("Cool blues")(4))
show_col(canva_pal("Modern and crisp")(4))
```

---

**canva_palettes**  
*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". They were then adapted to Tableau and Excel palettes by the sources below.

**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".

**Source**
References

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

Examples

```r
## Not run:
require("tidyverse")
canva_df <- map_df(canva_palette, names(canva_palette),
  - tibble(color = .x, .id = seq_along(color), palette = .y))
ggplot(canva_df, aes(y = palette, x = .id, fill = color)) +
geom_raster() +
scale_fill_identity(guide = FALSE) +
theme_minimal() +
theme(panel.grid= element_blank(),
  axis.text.x = element_blank()) +
labs(x = "", y = "")
## End(Not run)
```

circlefill_shape_pal  Filled Circle Shape palette (discrete)

Description

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.

Usage

circlefill_shape_pal()

References


See Also

Other shapes: cleveland_shape_pal, scale_shape_circlefill, scale_shape_cleveland, scale_shape_tremmel, tremmel_shape_pal
Examples

```r
library("ggplot2")
(ggplot(mtcars, aes(x=mpg, y=hp, shape=factor(cyl)))
  + geom_point() + scale_shape_tremmel())
```

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

Description

Shape palettes for overlapping and non-overlapping points.

Usage

```r
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 rely 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), I replace the circle with a vertical line with an encircled plus sign.

References


See Also

Other shapes: `circlefill_shape_pal`, `scale_shape_circlefill`, `scale_shape_cleveland`, `scale_shape_tremmel`, `tremmel_shape_pal`
Examples

```r
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)
```

colorblind_pal  
*Colorblind Color Palette (Discrete) and Scales*

Description

An 8-color colorblind safe qualitative discrete palette.

Usage

```r
colorblind_pal()

scale_colour_colorblind(...)

scale_color_colorblind(...)

scale_fill_colorblind(...)
```

Arguments

```r
...
```

Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

References

Chang, W. "Cookbook for R"

[http://jfly.iam.u-tokyo.ac.jp/color](http://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.

**Usage**

```r
economist_pal(stata = FALSE, fill = TRUE)
```

**Arguments**

- **stata**  
  Use the palette in the Stata economist scheme.

- **fill**  
  Use the fill palette.

**See Also**

Other colour economist: `scale_colour_economist`

**Examples**

```r
library(scales)
show_col(economist_pal)(6))
## fill palette
show_col(economist_pal(fill=TRUE)(6))
## RGB values from Stata's economist scheme
show_col(economist_pal(stata=TRUE)(16))
```
excel_pal

Description

Color palettes from Excel, both current and the pre-2007 ugly palettes.

Usage

excel_pal(palette = "line")

Arguments

palette One of 'old', 'fill', or 'new'.

Details

The color palettes are

- **line**  Excel 2003 default color palette.
- **fill**  Excel 2003 bar chart color palette.
- **new**  Color palette from newer Excel versions.

See Also

Other colour excel: scale_fill_excel

Examples

```r
library("scales")
show_col(excel_pal("line"))
show_col(excel_pal("fill"))
show_col(excel_pal("new"))
```

___

extended_range_breaks_

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.
extended_range_breaks_

Usage

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 Auguie

References

few_pal  

Color Palettes from Few’s "Practical Rules for Using Color in Charts"

Description
Qualitative color palettes from Stephen Few, "Practical Rules for Using Color in Charts".

Usage
```r
default_pal(palette = "medium")
```

Arguments
- **palette** One of "medium", "dark", or "light"

Details
He suggests the following
- For bars, use medium.
- For lines and points use dark if small or thin, and medium otherwise.

See Also
- Other colour few: `scale_colour_few`

Examples
```r
library("scales")
show_col(few_pal("medium"))
show_col(few_pal("dark"))
show_col(few_pal("light"))
```

fivethirtyeight_pal  
fivethirtyeight.com color palette

Description
The standard fivethirtyeight.com palette for line plots is blue, red, green.

Usage
```r
fivethirtyeight_pal()
```

See Also
- Other colour fivethirtyeight: `scale_colour_fivethirtyeight`
**gdocs_pal**

**Examples**

```r
library("scales")
show_col(fivethirtyeight_pal()(3))
```

---

**gdocs_pal**

*Google Docs color palette (discrete)*

**Description**

Color palettes from Google Docs.

**Usage**

```r
gdocs_pal()
```

**See Also**

Other colour gdocs: `scale_fill_gdocs`

**Examples**

```r
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**

```r
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` or `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.

stat  The statistical transformation to use on the data for this layer, as a string.

position  Position adjustment, either as a string, or the result of a call to a position adjustment function.

...  other arguments passed on to `layer`. These are often aesthetics, used to set an aesthetic to a fixed value, like `color = "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.

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`.

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() +
  theme_tufte()
```

---

### geom_tufteboxplot

**Tufte’s Box Blot**

**Description**

Edward Tufte’s revisions of the box plot as described in *The Visual Display of Quantitative Information*.

**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 `aes` or `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.

- `stat`: The statistical transformation to use on the data for this layer, as a string.

- `position`: Position adjustment, either as a string, or the result of a call to a position adjustment function.

- `outlier.colour`: Colour for outlying points

- `outlier.shape`: Shape of outlying points

- `outlier.size`: Size of outlying points

- `outlier.stroke`: Stroke for outlying points
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 \( y \).

\texttt{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 \( x \).

\texttt{na.rm} If \texttt{FALSE}, the default, missing values are removed with a warning. If \texttt{TRUE}, missing values are silently removed.

\texttt{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.

\texttt{inherit.aes} If \texttt{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. \texttt{borders}.

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

\texttt{whisker.type} If \texttt{'line'}, then whiskers are represented by lines. If \texttt{'point'}, then whiskers are represented by points at \( y_{\text{min}} \) and \( y_{\text{max}} \).

\texttt{...} other arguments passed on to \texttt{layer}. These are often aesthetics, used to set an aesthetic to a fixed value, like \texttt{color = "red"} or \texttt{size = 3}. They may also be parameters to the paired geom/stat.

\textbf{Aesthetics}

- \texttt{x} [required]
- \texttt{y} [required]
- \texttt{colour}
- \texttt{size}
- \texttt{linetype}
- \texttt{shape}
- \texttt{fill}
- \texttt{alpha}

\textbf{References}


\textbf{See Also}

\texttt{geom_boxplot}

Other geom tufte: \texttt{geom_rangeframe}
Examples

```r
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")
```

Description

This package contains extra themes, scales, and geoms, and functions for and related to `ggplot2`.

Details

In addition to the help pages, see the README page on github for examples.

```
ggthemes_data  Palette data for the gghthemes package
```

Description

Data used by the palettes in the gghthemes package.

Usage

`ggthemes_data`

Format

A list.
**Description**

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

**Usage**

```r
hc_pal(palette = "default")
```

**Arguments**

- **palette**: character The color palette to use. This must be a name in `ggthemes_data$hc$palettes`.

**Palettes**

The following palettes are defined,

- **default**: `#7cb5ec, #434348, #90ed7d, #f7a35c, #8085e9, #f15c80', #e4d354, #8085e8, #8d4653, #91e8e1` theme. Examples: [http://www.highcharts.com/demo](http://www.highcharts.com/demo).

- **darkunica**: `#2b908f, #90ee7e, #f45b5b, #7798BF, #aaeeee, #ff0066, #eeaaee, #55BF3B, #DF5353, #7798BF, #aaeeee'`. Examples: [http://www.highcharts.com/demo/line-basic/dark-una](http://www.highcharts.com/demo/line-basic/dark-una).

---

**palette_pander**

**Description**

The `pander` ships with a default colorblind and printer-friendly color palette borrowed from [http://jfly.iam.u-tokyo.ac.jp/color/](http://jfly.iam.u-tokyo.ac.jp/color/).

**Usage**

```r
palette_pander(n, random_order = FALSE)
```

**Arguments**

- **n**: number of 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`
ptol_pal

Examples

```r
## 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

[https://personal.sron.nl/~pault/colourschemes.pdf](https://personal.sron.nl/~pault/colourschemes.pdf)

See Also

Other colour ptol: `scale_colour_ptol`

Examples

```r
library("scales")
show_col(ptol_pal()(6))
show_col(ptol_pal()(4))
show_col(ptol_pal()(12))
```
**scale_color_pander**  
Color scale from the pander package

**Description**

The **pander** ships with a default colorblind and printer-friendly color palette borrowed from [http://jfly.iam.u-tokyo.ac.jp/color/](http://jfly.iam.u-tokyo.ac.jp/color/).

**Usage**

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

**Arguments**

```
...  Other arguments passed on to discrete_scale to control name, limits, breaks, labels and so forth.  
```

**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`.

**Usage**

```
scale_colour_canva(..., palette = "Fresh and bright")  
scale_color_canva(..., palette = "Fresh and bright")  
scale_fill_canva(..., palette = "Fresh and bright")  
```

**Arguments**

```
...  Arguments passed to discrete_scale.  
palette  Palette name. See the names of `canva_palettes` for valid names.  
```
**scale_colour_economist**

Economist color scales

### Description
Color scales using the colors in the Economist graphics.

### Usage
- `scale_colour_economist(stata = FALSE, ...)`
- `scale_color_economist(stata = FALSE, ...)`
- `scale_fill_economist(stata = FALSE, ...)`

### Arguments
- **stata**  
  Use the palette in the Stata economist scheme.
- **...**  
  Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

### See Also
- `theme_economist` for examples.
- Other colour economist: `economist_pal`

---

**scale_colour_few**  

Color scales from Few’s "Practical Rules for Using Color in Charts"

### Description
See `few_pal`.

### Usage
- `scale_colour_few(palette = "medium", ...)`
- `scale_color_few(palette = "medium", ...)`
- `scale_fill_few(palette = "light", ...)`
Arguments

palette One of "medium", "dark", or "light"
...

Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

See Also

Other colour few: `few_pal`

---

scale_colour_fivethirtyeight

*fivethirtyeight.com color scales*

---

Description

Color scales using the colors in the fivethirtyeight graphics.

Usage

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

Arguments

...

Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

See Also

`theme_fivethirtyeight` for examples.

Other colour fivethirtyeight: `fivethirtyeight_pal`
scale_colour_gradient2_tableau

Tableau diverging colour scales (continuous)

Description

Tableau diverging colour scales (continuous)

Usage

scale_colour_gradient2_tableau(palette = "Red-Blue", ..., space = "rgb", na.value = "grey50", guide = "colourbar")

scale_fill_gradient2_tableau(palette = "Red-Blue", ..., space = "rgb", na.value = "grey50", guide = "colourbar")

scale_color_gradient2_tableau(palette = "Red-Blue", ..., space = "rgb", na.value = "grey50", guide = "colourbar")

Arguments

palette

Palette name. See ggtrends_data$tableau$divergent.

... Other arguments passed on to discrete_scale to control name, limits, breaks, labels and so forth.

space

Colour space in which to calculate gradient.

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_div_gradient_pal, tableau_seq_gradient_pal

Examples

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))

p + scale_colour_gradient2_tableau()
p + scale_colour_gradient2_tableau('Orange-Blue')
p + scale_colour_gradient2_tableau('Temperature')
scale_colour_gradient_tableau

*Tableau sequential colour scale (continuous)*

**Description**

Tableau sequential colour scale (continuous)

**Usage**

```r
scale_colour_gradient_tableau(palette = "Red", ..., space = "Lab",
na.value = "grey50", guide = "colourbar")
```

```r
color_fill_gradient_tableau(palette = "Red", ..., space = "Lab",
na.value = "grey50", guide = "colourbar")
```

```r
color_gradient_color_tableau(palette = "Red", ..., space = "Lab",
na.value = "grey50", guide = "colourbar")
```

```r
color_gradient_continuous(palette = "Red", ..., space = "Lab",
na.value = "grey50", guide = "colourbar")
```

```r
color_fill_continuous(palette = "Red", ..., space = "Lab",
na.value = "grey50", guide = "colourbar")
```

**Arguments**

- `palette` Palette name. See `ggthemes_data$tableau$sequential`.
- `...` Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.
- `space` Colour space in which to calculate gradient.
- `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_div_gradient_pal, tableau_seq_gradient_pal`

**Examples**

```r
#
library("ggplot2")
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()

p + scale_colour_gradient_tableau("Red")
p + scale_colour_gradient_tableau("Blue")
p + scale_colour_gradient_tableau("Green")

---

**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 color palette to use. This must be a name in `ggthemes_data$hc$palettes`.
- `...` Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

---

**scale_colour_ptol**

*Color Scales from Paul Tol’s "Colour Schemes"

**Description**

See `ptol_pal`. 

scale_colour_stata

Usage

scale_colour_ptol(...)  
scale_color_ptol(...)  
scale_fill_ptol(...)

Arguments

... Other arguments passed on to discrete_scale to control name, limits, breaks, labels and so forth.

See Also

Other colour ptol: ptol_pal

scale_colour_stata  Stata color scales

Description

See stata_pal for details.

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".  
... Other arguments passed on to discrete_scale to control name, limits, breaks, labels and so forth.
scale_colour_tableau  Tableau color scales.

Description

See `tableau_color_pal` for details.

Usage

scale_colour_tableau(palette = "tableau10", ...)

scale_fill_tableau(palette = "tableau10", ...)

scale_color_tableau(palette = "tableau10", ...)

Arguments

- `palette`  Palette name.
- `...`  Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

See Also

- `tableau_color_pal` for references.

Other colour tableau: `scale_colour_gradient2_tableau`, `scale_colour_gradient_tableau`, `tableau_color_pal`, `tableau_div_gradient_pal`, `tableau_seq_gradient_pal`

Examples

```r
library("ggplot2")
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg, colour=factor(gear))) + facet_wrap(~am)
p + scale_colour_tableau()
p + scale_colour_tableau('tableau20')
p + scale_colour_tableau('tableau0medium')
p + scale_colour_tableau('tableau0light')
p + scale_colour_tableau('colorblind10')
p + scale_colour_tableau('trafficlight')
p + scale_colour_tableau('purplegray12')
p + scale_colour_tableau('bluered12')
p + scale_colour_tableau('greenorange12')
p + scale_colour_tableau('cyclic')
```
### scale_colour_wsj  
*Wall Street Journal color and fill scales*

**Description**

Colour and fill scales which use the palettes in `wsj_pal` and are meant for use with `theme_wsj`.

**Usage**

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

**Arguments**

- `palette` character: The color palette to use. This must be a name in `ggthemes_data$wsj$palettes`.
- `...` Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

**See Also**

Other colour wsj: `wsj_pal`

### scale_fill_calc  
*LibreOffice Calc color scales*

**Description**

Color scales from LibreOffice Calc.

**Usage**

```r
scale_fill_calc(...)  
scale_colour_calc(...)  
scale_color_calc(...)
```

**Arguments**

- `...` Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.
scale_fill_excel

See Also

See theme_calc for examples.
Other colour calc: calc_pal

scale_fill_excel  Excel color scales

Description

Color scales from both old and new Excel.

Usage

scale_fill_excel(palette = "fill", ...)
scale_colour_excel(palette = "line", ...)
scale_color_excel(palette = "line", ...)

Arguments

palette  One of 'old', 'fill', or 'new'.
...
Other arguments passed on to discrete_scale to control name, limits, breaks, labels and so forth.

See Also

See theme_excel for examples.
Other colour excel: excel_pal

scale_fill_gdocs  Google Docs color scales

Description

Color scales from Google Docs.

Usage

scale_fill_gdocs(...)
scale_colour_gdocs(...)
scale_color_gdocs(...)


Arguments

... Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

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

```
scale_fill_solarized(accen=“blue”, …)
scale_colour_solarized(accen=“blue”, …)
scale_color_solarized(accen=“blue”, …)
```

Arguments

accent character Starting color.
... Other arguments passed on to `discrete_scale` to control name, limits, breaks, labels and so forth.

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()
```
scale_linetype_stata  

Stata linetype palette (discrete)

Description

See `stata_linetype_pal` for details.

Usage

scale_linetype_stata(...)

Arguments

... common discrete scale parameters: name, breaks, labels, na.value, limits and guide. See `discrete_scale` for more details

See Also

Other linetype stata: `stata_linetype_pal`

Examples

library("reshape2") # for melt
library("plyr") # for ddply
library("ggplot2")
ecm <- melt(economics, id = "date")
rescale01 <- function(x) {(x - min(x)) / diff(range(x))}
ecm <- ddply(ecm, "variable", transform, value = rescale01(value))
ggplot(ecm, 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

scale_shape_calc(...)

Arguments

... common discrete scale parameters: name, breaks, labels, na.value, limits and guide. See `discrete_scale` for more details
scale_shape_circlefill

*Filled Circle Shape palette (discrete)*

---

**Description**

Filled Circle Shape palette (discrete)

**Usage**

```r
scale_shape_circlefill(...)```

**Arguments**

```
... common discrete scale parameters: name, breaks, labels, na.value, limits
and guide. See discrete_scale for more details
```

**See Also**

- `circlefill_shape_pal` for a description of the palette.
- 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**

```r
scale_shape_cleveland(overlap = TRUE, ...)
```

**Arguments**

```
overlap logical Use the scale for overlapping points?
... common discrete scale parameters: name, breaks, labels, na.value, limits
and guide. See discrete_scale for more details
```
scale_shape_stata

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_stata  Stata shape scale

description
See stata_shape_pal for details.

Usage
scale_shape_stata(...)  

Arguments
...  common discrete scale parameters: name, breaks, labels, na.value, limits and guide. See discrete_scale for more details

Examples
library("ggplot2")
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, shape = factor(gear))) +
  facet_wrap(~am)
p + theme_stata() + scale_shape_stata()

scale_shape_tableau  Tableau shape scales

description
See tableau_shape_pal for details.

Usage
scale_shape_tableau(palette = "default", ...)

scale_shape_tableau
scale_shape_tremmel

Arguments

palette          Palette name. See ggtthemes_data$tableau$shapes.

...              common discrete scale parameters: name, breaks, labels, na.value, limits and guide. See discrete_scale for more details

See Also

Other shape tableau: tableau_shape_pal

Examples

library("ggplot2")
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, shape = factor(gear))) +
  facet_wrap(~am)
p + scale_shape_tableau()

scale_shape_tremmel Shape scales from Tremmel (1995)

Description

Shape scales from Tremmel (1995)

Usage

scale_shape_tremmel(overlap = FALSE, n3alt = TRUE, ...)

Arguments

overlap          use an empty circle instead of a solid circle when n == 2.
n3alt            If TRUE then use a solid circle, plus sign and empty triangle, else use a solid circle, empty circle, and empty triangle.

...              common discrete scale parameters: name, breaks, labels, na.value, limits and guide. See discrete_scale for more details

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
show_linetypes

Examples

library("ggplot2")
(ggplot(mtcars, aes(x=mpg, y=hp, shape=factor(cyl)))
  + geom_point() + scale_shape_tremmel())
(ggplot(mtcars, aes(x=mpg, y=hp, shape=factor(cyl)))
  + geom_point() + scale_shape_tremmel(n3alt=FALSE))
(ggplot(mtcars, aes(x=mpg, y=hp, shape=factor(am)))
  + geom_point() + scale_shape_tremmel())
(ggplot(mtcars, aes(x=mpg, y=hp, shape=factor(am)))
  + geom_point() + scale_shape_tremmel(overlap=TRUE))

show_linetypes  Show linetypes

Description
A quick and dirty way to show linetypes.

Usage

show_linetypes(linetypes, labels = TRUE)

Arguments

linetypes  A character vector of linetypes. See par.
labels    Label each line with its linetype (ltY) value.

See Also

show_col, show_linetypes

Examples

library("scales")
show_linetypes(linetype_pal()(3))
show_linetypes(linetype_pal()(3), labels=TRUE)
show_shapes

Description
A quick and dirty way to show shapes.

Usage
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.

See Also
- show_col, show_linetypes

Examples
library("scales")
show_shapes(shape_pal()(5))
show_shapes(shape_pal()(3), labels=TRUE)

smart_digits

Description
Format numbers with automatic number of digits

Usage
smart_digits(x, ...)

Arguments
- x: A numeric vector to format
- ...: Parameters passed to format
solarized_pal

Value

smart_digits returns 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 Auguié, Jeffrey B. Arnold

References

23171858#23171858.

solarized_pal Solarized color palette (discrete)

Description


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 colour: 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.

**Usage**

stata_linetype_pal()

**See Also**

scale_linetype_stata

Other linetype stata: scale_linetype_stata

---

**stata_pal**

*Stata color palettes (discrete)*

**Description**


**Usage**

stata_pal(scheme = "s2color")

**Arguments**

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

**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_pal **  

*Stata shape palette (discrete)*

**Description**

Shape palette based on the symbol palette in Stata, specifically that for the scheme s2mono.

**Usage**

```r
stata_shape_pal()
```

**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", qs = 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` or `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. |
| geom    | The geometric object to use display the data |
| qs      | Quantiles to use for the five number summary. |
tableau_color_pal

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>na.rm</td>
<td>If FALSE (the default), removes missing values with a warning. If TRUE silently removes missing values.</td>
</tr>
<tr>
<td>position</td>
<td>Position adjustment, either as a string, or the result of a call to a position adjustment function.</td>
</tr>
<tr>
<td>show.legend</td>
<td>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.</td>
</tr>
<tr>
<td>inherit.aes</td>
<td>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.</td>
</tr>
<tr>
<td>...</td>
<td>other arguments passed on to layer. These are often aesthetics, used to set an aesthetic to a fixed value, like color = &quot;red&quot; or size = 3. They may also be parameters to the paired geom/stat.</td>
</tr>
</tbody>
</table>

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

Color Palettes based on Tableau (discrete)

Description

Color palettes used in Tableau.

Usage

tableau_color_pal(palette = "tableau10")

Arguments

- palette: Palette name.
References

http://vis.stanford.edu/color-names/analyzer/


See Also

Other colour tableau: `scale_colour_gradient2_tableau`, `scale_colour_gradient_tableau`, `scale_colour_tableau`, `tableau_div_gradient_pal`, `tableau_seq_gradient_pal`

Examples

```r
library("scales")
show_col(tableau_color_pal('tableau20')(20))
show_col(tableau_color_pal('tableau10')(10))
show_col(tableau_color_pal('tableau10medium')(10))
show_col(tableau_color_pal('tableau10light')(10))
show_col(tableau_color_pal('colorblind10')(10))
show_col(tableau_color_pal('trafficlight')(10))
show_col(tableau_color_pal('purplegray12')(12))
show_col(tableau_color_pal('bluered12')(12))
show_col(tableau_color_pal('greenorange12')(12))
show_col(tableau_color_pal('cyclic')(20))
```

---

date: 

title: Tableau diverging colour gradient palettes (continuous)

description: Tableau diverging colour gradient palettes (continuous)

Usage

`tableau_div_gradient_pal(palette = "Red-Blue", space = "Lab")`

Arguments

- **palette**: Palette name. See `ggthemes_data$tableau$divergent`.
- **space**: Colour space in which to calculate gradient.
See Also

Other colour tableau: `scale_colour_gradient2_tableau`, `scale_colour_gradient_tableau`, `scale_colour_tableau`, `tableau_color_pal`, `tableau_seq_gradient_pal`

Examples

```r
library(scales)
x <- seq(-1, 1, length = 100)
r <- sqrt(outer(x^2, x^2, '+'))
image(r,
   col = tableau_div_gradient_pal()$seq(0, 1, length = 12))
image(r,
   col = tableau_div_gradient_pal('Orange-Blue')$seq(0, 1, length = 12))
image(r,
   col = tableau_div_gradient_pal('Temperature')$seq(0, 1, length = 12))
```

tableau_seq_gradient_pal

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

Description

Tableau sequential colour gradient palettes (continuous)

Usage

```
tableau_seq_gradient_pal(palette = "Red", space = "Lab")
```

Arguments

- `palette` Palette name. See `ggthemes_data$tableau$sequential`.
- `space` Colour space in which to calculate gradient.

See Also

Other colour tableau: `scale_colour_gradient2_tableau`, `scale_colour_gradient_tableau`, `scale_colour_tableau`, `tableau_color_pal`, `tableau_seq_gradient_pal`

Examples

```r
library("scales")
x <- seq(0, 1, length = 25)
show_col(tableau_seq_gradient_pal('Red')(x))
show_col(tableau_seq_gradient_pal('Blue')(x))
show_col(tableau_seq_gradient_pal('Purple Sequential')(x))
```
tableau_shape_pal

Tableau Shape Palettes (discrete)

Description
Shape palettes used by Tableau.

Usage
tableau_shape_pal(palette = "default")

Arguments
palette Palette name. See ggthemes_data$tableau$shapes.

See Also
Other shape tableau: scale_shape_tableau

Examples
show_shapes(tableau_shape_pal()(5))

theme_base

Theme Base

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

Usage
theme_base(base_size = 16, base_family = "")

Arguments
base_size base font size
base_family base font family

See Also
Other themes: theme.Foundation, theme_igray, theme_par, theme_solid

Examples
library("ggplot2")
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg,
colour=factor(gear))) + facet_wrap(~am)
p + theme_base()
### theme_calc

**Theme Calc**

**Description**

Theme similar to the default settings of LibreOffice Calc charts.

**Usage**

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

**Arguments**

- `base_size` : base font size
- `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) + 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()
```

### theme_economist

**ggplot color theme based on the Economist**

**Description**

Style plots similar to those in *The Economist*.

**Usage**

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

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

Arguments

- base_size: base font size
- base_family: base font family
- horizontal: logical. Horizontal axis lines?
- dkpanel: logical. Darker background for panel region?
- stata: logical. Use RGB values from Stata's economist scheme.
- 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 used by *The Economist* blog Graphic Detail.

*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

See Also

theEconomist.theme for an Economist theme for lattice plots.

Examples

```r
library("ggplot2")
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg, colour=factor(gear))) + facet_wrap(~am) +
   # Economist puts scales on the right-hand side
   scale_y_continuous(position = "right")

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

## Stata colors
p + theme_economist(stata=TRUE) + scale_colour_economist(stata=TRUE)

## Darker plot region
p + theme_economist(dkpanel=TRUE) + scale_colour_economist(stata=TRUE)
```
# 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 color 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.

**Usage**

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

**Arguments**

- `base_size`  
  base font size

- `base_family`  
  base font family

- `horizontal`  
  logical. Horizontal axis lines?

**Value**

An object of class `theme`. 
theme_few

Examples

library("ggplot2")
# Old line color
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg, colour=factor(gear))) +
  facet_wrap(~am)
p + theme_excel() + scale_colour_excel()

# Old fill color palette
ggplot(mpg, aes(x = class, fill = drv)) +
  geom_bar() +
  scale_fill_excel('fill') +
  theme_excel()

theme_few

Theme based on Few’s "Practical Rules for Using Color in Charts"

Description

Theme based on the rules and examples in Stephen Few, "Practical Rules for Using Color in Charts"

Usage

theme_few(base_size = 12, base_family = "")

Arguments

base_size base font size
base_family base font family

References


Examples

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")
theme_fivethirtyeight  Theme inspired by fivethirtyeight.com plots

Description
Theme inspired by the plots on http://fivethirtyeight.com.

Usage
theme_fivethirtyeight(base_size = 12, base_family = "sans")

Arguments
base_size  base font size
base_family  base font family

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

desc_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
theme_foundation(base_size = 12, base_family = "")

Arguments
base_size  base font size
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_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

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

Arguments

- base_size  base font size
- base_family  base font family

Examples

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 JS theme

Description

Theme based on the plots in Highcharts JS.

Usage

theme_hc(base_size = 12, base_family = "sans", bgcolor = "default")
theme_igray

Arguments

base_size  base font size
base_family base font family
bgcolor    The background color of plot. One of 'default', 'darkunica', the names of values in ggthemes_data$hc$bg.

References

http://www.highcharts.com/demo/line-basic
https://github.com/highslide-software/highcharts.com/tree/master/js/themes

Examples

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')

theme_igray  Inverse gray theme

Description

Theme with white panel and gray background.

Usage

theme_igray(base_size = 12, base_family = "")

Arguments

base_size  base font size
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.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
- `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, eval.mgs or Pandoc.brew.

Usage

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

library("ggplot2")
library("pander")

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

panderoptions('graph.grid.color', 'red')
**theme_par**

Theme which takes its values from the current ‘base’ graphics parameter values in `par`.

**Description**


**Usage**

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

**Arguments**

- `base_size` base font size
- `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_foundation`, `theme_igray`, `theme_solid`

**Examples**

```
library("ggplot2")
p <- ggplot(mtcars) + geom_point(aes(x = wt, y = mpg,
  colour=factor(gear))) + facet_wrap(~am)
par(font = 2, col.lab = "red", fg = "blue")
p + theme_par()
```
theme_solarized  ggplot color themes based on the Solarized palette

Description

Usage
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
- **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/beamercolorthemessolarized](https://github.com/jrnold/beamercolorthemessolarized). There are two variations: `theme_solarized` is similar to `theme_bw`, while `theme_solarized_2` is similar to `theme_gray`.

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('blue')

# Dark version
p + theme_solarized(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.
theme_stata

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_foundation`, `theme_igray`, `theme_par`

Examples

```
library("ggplot2")
(ggplot(mtcars, aes(wt, mpg))
 + geom_point()
 + theme_solid(fill = "white"))
```

---

**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")
```

Arguments

- **base_size**: Base font size
- **base_family**: Base font family
- **scheme**: One of "s2color", "s2mono", "s1color", "s1rcolor", or "s1mono", "s2manual", "s1manual", or "sj"

Note

Stata graph schemes include the features of `ggplot2` into themes and scales. Stata graph themes also allow for defaults for specific graph types, a feature which `ggplot2` does not directly support.

References

Examples

```r
library("ggplot2")
p <- ggplot(mtcars) +
geom_point(aes(x = wt, y = mpg, colour=factor(gear))) +
facet_wrap(~am)
# s2color
p + theme_stata() + scale_colour_stata("s2color")
# s2mono
p + theme_stata(scheme = "s2mono") + scale_colour_stata("mono")
# s1color
p + theme_stata(scheme = "s1color") + 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  
**Tuft 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
- `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")
# with ticks and range frames
(ggplot(mtcars, aes(x, y))
 + geom_point() + geom_rangeframe()
 + theme_tufte())

# with geom_rug
(ggplot(mtcars, aes(x, y))
 + geom_point() + geom_rug()
 + theme_tufte(ticks=FALSE))

## Not run:
## Using the Bembo serif family
library(extrafont)
(ggplot(mtcars, aes(x, y))
 + geom_point() + geom_rangeframe()
 + theme_tufte(base_family='BemboStd'))

## Using the Gill Sans sans serif family
(ggplot(mtcars, aes(x, y))
 + geom_point() + geom_rangeframe()
 + theme_tufte(base_family='GillSans'))

## End(Not run)
```

---

**theme_wsj**  
Wall Street Journal theme

---

**Description**  
Theme based on the plots in *The Wall Street Journal.*

**Usage**

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

**Arguments**

- **base_size**: base font size
- **color**: The background color of plot. One of 'brown', 'gray', 'green', 'blue', the names of values in ggthemes_data$wsj$bg.
- **base_family**: base font family
- **title_family**: Plot title font family.

**References**

- [https://twitter.com/WSJGraphics](https://twitter.com/WSJGraphics)
- [https://pinterest.com/wsjgraphics/wsj-graphics/](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:

Usage

tremmel_shape_pal(overlap = FALSE, n3alt = TRUE)

Arguments

overlap  use an empty circle instead of a solid circle when n == 2.
n3alt    If TRUE then use a solid circle, plus sign and empty triangle, else use a solid circle, empty circle, and empty triangle.

Details

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).

If more than three groups of data, then separate the groups into different plots.

References


See Also

Other shapes: circlefill_shape_pal, cleveland_shape_pal, scale_shape_circlefill, scale_shape_cleveland, scale_shape_tremmel
**wsj_pal**

*Wall Street Journal color palette (discrete)*

**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 can be found on the WSJ Graphics Twitter feed and Pinterest.

**Usage**

```r
wsj_pal(palette = "colors6")
```

**Arguments**

- `palette` character The color palette to use. This must be a name in `ggthemes_data$wsj$palettes`.

**Palettes**

The following palettes are defined,

- **rgby** Red/Green/Blue/Yellow theme. Examples: [https://twitpic.com/b2e3v2](https://twitpic.com/b2e3v2).
- **green_black** Black-green 4-color scale for 'Very negative', 'Somewhat negative', 'somewhat positive', 'very positive'. Examples: [https://twitpic.com/awbua0](https://twitpic.com/awbua0).
- **dem_rep** Democrat/Republican/Undecided blue/red/gray scale. Examples: [https://twitpic.com/awbua0](https://twitpic.com/awbua0).
- **colors6** Red, blue, gold, green, orange, and black palette. Examples: [https://twitpic.com/9gfg5q](https://twitpic.com/9gfg5q).

**See Also**

Other colour wsj: `scale_colour_wsj`
Index

*Topic datasets
  - canva_palettes, 7
  - geom_rangeframe, 15
  - geom_tufteboxplot, 17
  - ggthemes_data, 19
  - stat_fivenumber, 41
  - aes, 16, 17, 41
  - aes_, 16, 17, 41

  - bank_slopes, 4
  - banking, 5
  - borders, 16, 18, 42

  - calc_pal, 6, 31
  - calc_shape_pal, 6, 33, 34
  - canva_pal, 7
  - canva_palettes, 7, 7, 22
  - circlefill_shape_pal, 8, 9, 34–36, 60
  - cleveland_shape_pal, 8, 9, 34–36, 60
  - colorblind_pal, 10

  - dichromat_pal, 10
  - discrete_scale, 10, 22–36

  - economist_pal, 11, 23
  - excel_pal, 12, 31
  - extended_range_breaks
    - (extended_range_breaks_), 12
  - extended_range_breaks__, 12

  - few_pal, 14, 23, 24
  - fivethirtyeight_pal, 14, 24
  - format, 38
  - fortify, 16, 17, 41

  - gdocs_pal, 15, 32
  - geom_boxplot, 18
  - geom_map, 53
  - geom_rangeframe, 15, 18, 58
  - geom_rug, 58

  - geom_tufteboxplot, 16, 17
  - geomRangeFrame (geom_rangeframe), 15
  - geomTufteboxplot (geom_tufteboxplot), 17
  - ggplot, 16, 17, 41
  - ggthemes, 19
  - ggthemes-package (ggthemes), 19
  - ggthemes_data, 19
  - ggthemes_data$hccpalettes, 20, 27
  - ggthemes_data$wsjpalettes, 30, 61

  - hc_pal, 20, 27

  - layer, 16, 18, 42

  - palette_pander, 20, 22
  - par, 37, 38, 55
  - ptol_pal, 21, 27, 28

  - scale_color_calc (scale_fill_calc), 30
  - scale_color_canva (scale_colour_canva), 22
  - scale_color_colorblind
    - (colorblind_pal), 10
  - scale_color_continuous_tableau
    - (scale_colour_gradient_tableau), 26
  - scale_color_economist
    - (scale_colour_economist), 23
  - scale_color_excel (scale_fill_excel), 31
  - scale_color_few (scale_colour_few), 23
  - scale_color_fivethirtyeight
    - (scale_colour_fivethirtyeight), 24
  - scale_color_gdocs (scale_fill_gdocs), 31
  - scale_color_gradient2_tableau
    - (scale_colour_gradient2_tableau), 25
  - scale_color_gradient_tableau
    - (scale_colour_gradient_tableau), 26
INDEX

scale_color_hc (scale_colour_hc), 27
scale_color_pander, 20, 22
scale_color_ptol (scale_colour_ptol), 27
scale_color_solarized
  (scale_fill_solarized), 32
scale_color_stata (scale_colour_stata), 28
scale_color_tableau, 10
scale_color_tableau
  (scale_colour_tableau), 29
scale_color_wsj (scale_colour_wsj), 30
scale_colour_calc (scale_fill_calc), 30
scale_colour_canva, 22
scale_colour_colorblind
  (colorblind_pal), 10
scale_colour_economist, 11, 23
scale_colour_excel (scale_fill_excel), 31
scale_colour_few, 14, 23
scale_colour_fivethirtyeight, 14, 24
scale_colour_gdocs (scale_fill_gdocs), 31
scale_colour_gradient2_tableau, 25, 26, 29, 43, 44
scale_colour_gradient_tableau, 25, 26, 29, 43, 44
scale_colour_hc, 27
scale_colour_pander
  (scale_color_pander), 22
scale_colour_ptol, 21, 27
scale_colour_solarized
  (scale_fill_solarized), 32
scale_colour_stata, 28
scale_colour_tableau, 25, 26, 29, 43, 44
scale_colour_wsj, 30, 61
scale_fill_calc, 6, 30
scale_fill_canva (scale_colour_canva), 22
scale_fill_colorblind (colorblind_pal), 10
scale_fill_continuous_tableau
  (scale_colour_gradient_tableau), 26
scale_fill_economist
  (scale_colour_economist), 23
scale_fill_excel, 12, 31
scale_fill_few (scale_colour_few), 23
scale_fill_fivethirtyeight
  (scale_colour_fivethirtyeight), 24
scale_fill_gdocs, 15, 31
scale_fill_gradient2_tableau
  (scale_colour_gradient2_tableau), 25
scale_fill_gradient_tableau
  (scale_colour_gradient_tableau), 26
scale_fill_hc (scale_colour_hc), 27
scale_fill_pander (scale_color_pander), 22
scale_fill_ptol (scale_colour_ptol), 27
scale_fill_solarized, 32, 39
scale_fill_stata (scale_colour_stata), 28
scale_fill_tableau
  (scale_colour_tableau), 29
scale_fill_wsj (scale_colour_wsj), 30
scale_linetype_stata, 33, 40
scale_shape_calc, 6, 33
scale_shape_cleveland, 8, 9, 34, 35, 36, 60
scale_shape_cleveland, 8, 9, 34, 36, 60
scale_shape_circlefill, 8, 9, 34, 35, 36, 60
scale_shape_stata, 35, 41
scale_shape_tableau, 35, 45
scale_shape_tremmel, 8, 9, 34, 35, 36, 60
show_col, 37, 38
show_linetypes, 37, 37, 38
show_shapes, 38
smart_digits, 38
smart_digits_format (smart_digits), 38
solarized_pal, 32, 39
stat_boxplot, 42
stat_fivenumber, 41
stata_linetype_pal, 33, 40
stata_pal, 28, 40
stata_shape_pal, 35, 41
StatFivenumber (stat_fivenumber), 41
tableau_color_pal, 25, 26, 29, 42, 44
tableau_div_gradient_pal, 25, 26, 29, 43, 44
tableau_seq_gradient_pal, 25, 26, 29, 43, 44, 44
tableau_shape_pal, 35, 36, 45
theEconomist.theme, 47
theme, 47, 48
theme_base, 45, 51, 52, 55, 57
theme_bw, 32, 36
theme_calc, 31, 34, 46
theme_economist, 23, 46
theme_economist_white
  (theme_economist), 46
theme_excel, 31, 48
theme_few, 49
theme_fivethirtyeight, 24, 50
theme.Foundation, 45, 50, 52, 55, 57
theme.gdocs, 32, 51
theme_gray, 52, 56
theme hc, 27, 51
theme_igray, 45, 51, 52, 55, 57
theme_map, 53
theme_pander, 22, 54
theme_par, 45, 51, 52, 55, 57
theme_solarized, 56
theme_solarized_2 (theme_solarized), 56
theme_solid, 45, 51, 52, 55, 56
theme_stata, 57
theme_tufte, 58
theme_wsj, 30, 59
tremmel_shape_pal, 8, 9, 34–36, 60

wsj_pal, 30, 61