

Package: stevethemes (via r-universe)

October 18, 2024

Type Package

Title Steve's 'ggplot2' Themes and Related Theme Elements

Depends R (>= 3.5.0)

Version 0.1.9

Maintainer Steve Miller <steven.v.miller@gmail.com>

Description This is a compilation of my preferred themes and related theme elements for 'ggplot2'. I believe these themes and theme elements are aesthetically pleasing, both for pedagogical instruction and for the presentation of applied statistical research to a wide audience. These themes imply routine use of easily obtained/free fonts, simple forms of which are included in this package.

URL <http://svmiller.com/stevethemes/>

BugReports <https://github.com/svmiller/stevethemes/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports ggplot2 (>= 3.3.0), rlang (>= 1.0.0), systemfonts

Repository <https://svmiller.r-universe.dev>

RemoteUrl <https://github.com/svmiller/stevethemes>

RemoteRef HEAD

RemoteSha 721be659ada166b9ac6f0bdcfd62d7b12a48a7ff

Contents

adj_font_size	2
are_fonts_installed	3
example_plot	4
g_c	5

how_to_install_fonts	5
kwh_gdp	6
martel_ratings	7
no_gridlines	7
se_counties_gdppc	8
steve_hex	9
theme_steve	10

Index	12
--------------	-----------

adj_font_size	<i>Miscellaneous functions for fine-tuning ggplot2 plots</i>
---------------	---

Description

These are assorted functions that will fine-tune various things in a **ggplot2** plot.

Usage

```
adj_font_size(x, ...)
```

Arguments

x	a numeric vector that will adjust the size of all fonts on the plot. Can take decimals and negative numbers.
...	optional, called for side effects

Details

adj_font_size() is a simple wrapper for some basic **ggplot2** code that will allow you to adjust the font size on the plot by some number.

Right now, adj_font_size() assumes you are using theme_steve() because the font sizes initialized in the function are defaults from the theme.

Value

adj_font_size() takes a plot made in **ggplot2** and changes the font size by some number requested by the user. It returns a plot with bigger or smaller fonts, per the user's request.

Author(s)

Steven V. Miller

Examples

```
library(ggplot2)

example_plot(type = "scatter") + adj_font_size(-2)

example_plot(type = "scatter") + adj_font_size(2)
```

are_fonts_installed *Check if you have the fonts you need already installed on your computer*

Description

are_fonts_installed() leans on **systemfonts** to determine if R can find the fonts that are used by the **stevethemes** package.

Usage

```
are_fonts_installed(check = "all")
```

Arguments

check a character vector coinciding with the font or fonts you want to check. Defaults to "all", which will check for all of "Open Sans", "Titillium Web", "Fira Sans", "Crimson Pro", and "Merriweather".

Details

Future functionality will allow you to supply more than "all", but "all" is the only argument for the function that will work for the moment.

Value

are_fonts_installed() returns a message communicating whether the **systemfonts** package in R can find the fonts used by **stevethemes** on the user's operating system. It also returns some information about where the user can download or obtain these fonts.

Author(s)

Steven V. Miller

Examples

```
are_fonts_installed()
```

example_plot	<i>Create an example plot (in ggplot2) for experimenting with different styles</i>
--------------	---

Description

example_plot() allows you to experiment with some **ggplot2** themes, like those in this package, by seeing them applied to some example plots.

Usage

```
example_plot(type = "bar")
```

Arguments

type a type

Details

I'll add more here in a little bit.

Value

example_plot() returns a plot made in **ggplot2**, allowing the user to experiment with different themes to see which ones they like the best.

Author(s)

Steven V. Miller

Examples

```
library(ggplot2)

example_plot()

example_plot(type = "faceted_line")

example_plot(type = "scatter")
```

`g_c`*Get a Custom Color from steve_hex*

Description

`g_c()` (i.e. **g*et *c*olor*) is a simple function that interfaces with the `steve_hex` data frame to return a hex value associated with a named color.

Usage

```
g_c(x)
```

Arguments

`x` a label coinciding with a value in the `color` column of the `steve_hex` data frame

Details

Check `steve_hex` in this same package for the custom colors included

Value

`g_c()` returns a character vector, in particular, a hex triplet that coincides with the label supplied in the function. Use it for specifying a custom fill or color in a plot.

Author(s)

Steven V. Miller

Examples

```
g_c("su_blue")
```

```
g_c("martel_pink")
```

`how_to_install_fonts` *Learn how to install fonts on your computer*

Description

`how_to_install_fonts()` communicates to the user how they can install fonts on their computer.

Usage

```
how_to_install_fonts(os = "any")
```

Arguments

`os` a character vector coinciding with the information you want, contingent on your operating system. If `os` is "any" (the default), it returns a generic message about how to install fonts. If `os` is "windows", "mac", or "linux", you get more information about font installation on your operating system.

Details

The 'any' option for the `os` argument is necessarily less informative. If you know whether you're using Windows, a Mac, or Linux, you should change the `os` argument.

Value

`how_to_install_fonts()` returns a message communicating how to install fonts on the user's operating system.

Author(s)

Steven V. Miller

Examples

```
how_to_install_fonts("windows")
```

```
how_to_install_fonts("mac")
```

kwh_gdp

Kilowatt Hours per Capita and GDP per Capita, 2010

Description

This is a simple data frame of the GDP per capita and kilowatt hours consumed per capita of over 130 sovereign states in 2010.

Usage

```
kwh_gdp
```

Format

A data frame the following 7 variables.

`country` the name of the country

`iso3c` the three-character ISO code of the country

`year` the year of observation (2010)

`kwhpc` the electric power consumption (kilowatt hours) per capita of the country

`gdppc` the GDP per capita of the country in current USD

ln_kwhpc the (log-transformed) electric power consumption (kilowatt hours) per capita of the country

ln_gdppc the (log-transformed) GDP per capita of the country in current USD

Details

Data come from a use of the `WDI()` function in the **WDI** package. The GDP per capita data come from a combination of the World Bank and OECD. The energy consumption data come from the International Energy Agency. Data exist to be used in a simple scatterplot.

martel_ratings	<i>CAGEMATCH Ratings of Rick Martel</i>
----------------	---

Description

This is a simple data frame of ratings (on a 1-10 scale) of Rick Martel.

Usage

```
martel_ratings
```

Format

A data frame the following 2 variables.

date a date vector for when the rating was posted on CAGEMATCH

value an individual person's rating of Rick Martel (on a 1-10 scale)

Details

Data exist to be used an example bar chart. CAGEMATCH (stylized in all caps) is an internet wrestling database, for which these ratings are fan submissions. Rick Martel is an objective 10/10.

no_gridlines	<i>Miscellaneous functions to make your ggplot2 plot "print-ready"</i>
--------------	---

Description

These are assorted functions that will make your plot "print-ready" by removing gridlines and giving hard-line axes to the plot. These are typically changes requested by publishes for the printing process.

Usage

```
no_gridlines(...)
```

```
make_classic(...)
```

Arguments

... optional, called for side effects

Details

`no_gridlines()` is a simple wrapper for some basic **ggplot2** code that will allow you to remove gridlines from the plot. `make_classic()` removes gridlines, removes the default gray background, *and* imposes a black, solid line on both axes. It will also hard code the axis text to be black. Using the latter with the former is likely redundant.

You will want to put these functions after a theme you've declared. If you run this before adding a theme over it, the theme you add will probably overwrite this function.

Value

`no_gridlines()` takes a plot made in **ggplot2** and removes the gridlines from the plot before returning it to the user. `make_classic()` takes a plot made in **ggplot2** and removes the gridlines, gray background, and adds solid axes to the plot.

Author(s)

Steven V. Miller

Examples

```
library(ggplot2)

example_plot(type = "scatter")

example_plot(type='scatter') + no_gridlines()

example_plot(type='scatter') + make_classic()
```

se_counties_gdppc	<i>GDP per Capita of Swedish Counties, 2001-2020</i>
-------------------	--

Description

This is a simple data frame of the GDP per capita of Swedish counties from 2001 to 2020.

Usage

```
se_counties_gdppc
```


Format

A data frame the following 4 variables.

nuts the Nomenclature of Territorial Units for Statistics (NUTS) code for the county

county the name of the county

year the year of observation

value the GDP per capita of the county in nominal SEK

Details

Data come from the OECD and exist to be used as a faceted line chart.

steve_hex *Some Hex Triplets I Find Useful/Interesting/Fun*

Description

This is a simple data frame with labels corresponding to hex triplets (i.e. web colors) that I find useful, interesting, or fun.

Usage

steve_hex

Format

A data frame the following 2 variables.

color a character vector describing the color in question

hex a hex triplet (with preceding hashtag) of the color

Details

Data exist to be used by `g_c()` in this same package.

`theme_steve`*Steve's Preferred ggplot2 Theme*

Description

`theme_steve()` is my default theme framework for graphs I make with **ggplot2**. It starts `theme_bw()`, which is available in **ggplot2**, but adjusts the margins and axes a bit to my liking. The end result is, I think, a lovely template for graphs I make in R.

Usage

```
theme_steve(style = "web", font, ...)
```

Arguments

<code>style</code>	various styles/adjustments to make to the base theme. Must be one of the following: "web" (default), "ms", "fira", "merri", "custom", or "generic". The "custom" style is a fancy way of saying "supply your own fonts". Where "custom" is used in this argument, something must be supplied to the font argument in this same function.
<code>font</code>	a character vector corresponding with a font that the user ideally has installed on their operating system.
<code>...</code>	optional, called for side effects

Details

The best use of this function may involve the fonts you have installed on your system. The user should experiment with various options to see what they like. The "generic" style will use default **ggplot2** fonts and is invoked here in this simple function for CRAN testing requirements. The default is "web", which depends on having Open Sans and Titillium Web on the operating system. You will see these two fonts everywhere on my website and most of my R package websites.

For greater clarity, the user should understand "ms" is code for the Crimson Pro font, which resembles the "cochineal" font family in the LaTeX font catalogue. "cochineal" is a fork of the Crimson font and its use is heavily implied by my preferred article template ('article2') in **stevetemplates**.

"fira" is shorthand for "Fira Sans". "merri" is shorthand for "Merriweather". The former features prominently in Firefox as well as my current operating system (Pop! OS). The latter incorporates the Merriweather font, which is a close cousin of the Palatino font. Palatino is the implied default font of my first article template ('article') in **stevetemplates**.

Value

No return value. Function is used for its side effect, which is to format a plot made in the **ggplot2** package.

Examples

```
library(ggplot2)

# Experiment with options, those this depends on fonts you have installed.
example_plot() + theme_steve(style='generic')
```

Index

* datasets

- kwh_gdp, [6](#)
- martel_ratings, [7](#)
- se_counties_gdppc, [8](#)
- steve_hex, [9](#)

adj_font_size, [2](#)

are_fonts_installed, [3](#)

example_plot, [4](#)

g_c, [5](#)

how_to_install_fonts, [5](#)

kwh_gdp, [6](#)

make_classic(no_gridlines), [7](#)

martel_ratings, [7](#)

no_gridlines, [7](#)

se_counties_gdppc, [8](#)

steve_hex, [9](#)

theme_steve, [10](#)