Package: hrbrthemes (via r-universe)

May 19, 2026

Type Package

Title Additional Themes, Theme Components and Utilities for 'ggplot2'

Version 0.9.2

Date 2025-03-14

Maintainer Bob Rudis <bob@rud.is>

Description A compilation of extra 'ggplot2' themes, scales and utilities, including a spell check function for plot label fields and an overall emphasis on typography. A copy of the 'Google' font 'Roboto Condensed' is also included.

Author @Author@

Encoding UTF-8

Suggests testthat, dplyr, gridExtra, hunspell, stringi, gcookbook, clipr, vdiffr, svglite

Depends R (>= 4.0.0)

Imports ggplot2 (>= 3.5.2), grDevices, grid, scales, tools, magrittr, utils

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Repository https://jd2112.r-universe.dev

Date/Publication 2026-04-19 13:43:31 UTC

RemoteUrl https://github.com/hrbrmstr/hrbrthemes

RemoteRef HEAD

RemoteSha 45ac19c58dc97079fb99e4277fd67993364ba005

Contents

`bit12` *Bit12 Color Palette*

Description

A vector of 12 colors in the Bit12 color scheme

Usage

```
bit12
```

Format

A named character vector of hex color codes

`bit12_extended` *Extended Bit12 Color Palettes*

Description

A list of 12 color palettes, each containing 9 shades for continuous color interpolation

Usage

```
bit12_extended
```

Format

A list of character vectors containing hex color codes

bit12_scale *Bit12 Color Palettes and Scales*

Description

Color palettes and scales based on the Bit12 color scheme, a vibrant 12-color system designed for clear data visualization. The palette offers a complete spectrum of distinct colors that work well for categorical data while maintaining good contrast and accessibility.

Usage

```
scale_color_bit12(...)

scale_fill_bit12(...)

scale_color_bit12_continuous(palette = "azure", ...)

scale_fill_bit12_continuous(palette = "azure", ...)

scale_color_bit12_distiller(palette = "azure", direction = 1, ...)

scale_fill_bit12_distiller(palette = "azure", direction = 1, ...)

scale_color_bit12_spectrum(direction = 1, ...)

scale_fill_bit12_spectrum(direction = 1, ...)
```

Arguments

...	Additional arguments passed to scale_color_manual(), scale_fill_manual(), scale_color_gradientn(), or scale_fill_gradientn()
palette	For continuous scales, the name of the color palette to use (one of "plum", "rose", "coral", "apricot", "lemon", "lime", "mint", "teal", "sky", "azure", "cobalt", "violet")
direction	Sets the direction of the color scale (1 = default, -1 = reversed)

Details

The package provides several types of scales:

- Discrete scales for categorical data
- Continuous scales with extended color ramps

The color palette includes:

- Plum: Deep purple for emphasis
- Rose: Rich pink-red for highlighting important elements

- Coral: Soft reddish-orange for warm secondary elements
- Apricot: Muted orange for highlighting
- Lemon: Bright yellow for high visibility elements
- Lime: Fresh green for positive values and growth
- Mint: Cool green-blue for calm elements
- Teal: Balanced blue-green for neutral data
- Sky: Light blue for information elements
- Azure: Clear blue for primary interactions
- Cobalt: Deep blue for stability and reliability
- Violet: Purple for creative and distinctive elements

Value

A ggplot2 scale object

Examples

```
library(ggplot2)

# Discrete color scale
ggplot(mpg, aes(class, fill = class)) +
  geom_bar() +
  scale_fill_bit12()

# Continuous color scale
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_bit12_continuous(palette = "azure")
```

`flexoki` *Flexoki Color Palettes and Scales*

Description

Color palettes and scales based on the Flexoki color scheme, a warm and soft color system designed by Steph Ango. Flexoki emphasizes readability and natural color harmony, making it particularly suitable for data visualization and long-form reading. The palette includes both light and dark variants, with each offering 8 distinct colors.

The name "Flexoki" combines "flexible" and "karaoke", reflecting its adaptable nature and harmonious color relationships.

Arguments

`...`	Additional arguments passed to scale_color_manual() or scale_fill_manual()
`palette`	For continuous scales, the name of the color palette to use (one of "red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")

Details

The package provides several types of scales:

- Discrete scales for both light and dark variants
- Continuous scales with extended color ramps

The color palette includes:

- Red: For emphasis and warnings
- Orange: For secondary emphasis
- Yellow: For highlighting
- Green: For success states and nature-related data
- Cyan: For information and water-related data
- Blue: For primary elements and cool tones
- Purple: For luxury and depth
- Magenta: For creative and feminine contexts

Value

A ggplot2 scale object

References

Flexoki Color System: https://stephango.com/flexoki

Examples

```
library(ggplot2)

# Discrete color scale (light variant)
ggplot(mpg, aes(class, fill = class)) +
  geom_bar() +
  scale_fill_flexoki_light()

# Continuous color scale
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_flexoki_continuous(palette = "blue")
```

flexoki_dark *Flexoki Dark Color Palette*

Description

A vector of 8 colors in the dark variant of the Flexoki color scheme

Usage

```
flexoki_dark
```

Format

A named character vector of hex color codes

flexoki_extended *Extended Flexoki Color Palettes*

Description

A list of 8 color palettes, each containing 13 shades for continuous color interpolation

Usage

```
flexoki_extended
```

Format

A list of character vectors containing hex color codes

flexoki_light *Flexoki Light Color Palette*

Description

A vector of 8 colors in the light variant of the Flexoki color scheme

Usage

```
flexoki_light
```

Format

A named character vector of hex color codes

flush_ticks *Makes axis text labels flush on the ends*

Description

A covenience function intended for basic, fixed-scale plots only (i.e. does not handle free scales in facets).

You need to pass in a ggplot2 object to this function. It can't be +'d in a chain of geoms, coords, scales, themes, etc. It also builds the plot (but does not display it) so if the plt takes a while (i.e. has lots of data or transforms) this will also take a while.

Usage

```
flush_ticks(gg, flush = "XY", plot = TRUE, cat = TRUE)
```

Arguments

gg	ggplot2 plot object
flush	either "X" or "Y" or "XY" to flush individual or both axes. Default: both.
plot	if FALSE then the ggplot object will be returned *invisibly*
cat	if TRUE then display theme() statements and copy them to the clipboard

Value

ggplot2 object with theme() elements added

Note

Intended for basic, fixed-scale plots only (i.e. does not handle free scales in facets).

font_an *Arial Narrow font name R variable aliases*

Description

font_an == "Arial Narrow"

Usage

```
font_an
```

Format

length 1 character vector

font_es *Econ Sans Condensed font name R variable aliases*

Description

font_es == "EconSansCndLig"

font_es_bold == "EconSansCndBol"

font_es_light == "EconSansCndLig"

Usage

```
font_es

font_es_bold

font_es_light
```

Format

length 1 character vector

An object of class character of length 1.

An object of class character of length 1.

Note

font_es_bold (a.k.a. "EconSansCndBol") is not available on Windows and will throw a warning if used in plots.

font_es_light (a.k.a. "EconSansCndLig") is not available on Windows and will throw a warning if used in plots.

font_gs *Goldman Sans font name R variable aliases*

Description

font_gs == "Goldman Sans Condensed"

Usage

```
font_gs
```

Format

length 1 character vector

font_inter_thin *Inter font name R variable aliases*

Description

font_inter_thin == "Inter-Thin"

font_inter_thin_italic == "Inter-ThinItalic"

font_inter_medium == "Inter-Medium"

font_inter_medium_italic == "Inter-MediumItalic"

font_inter_semibold == "Inter-SemiBold"

font_inter_bold == "Inter-Bold"

font_inter_italic == "Inter-Italic"

Usage

```
font_inter_thin

font_inter_thin_italic

font_inter_medium

font_inter_medium_italic

font_inter_semibold

font_inter_bold

font_inter_italic
```

Format

length 1 character vector

length 1 character vector

length 1 character vector

length 1 character vector

length 1 character vector

length 1 character vector

length 1 character vector

font_ps *PlexSans font name R variable aliases*

Description

`font_ps` == "IBMPlexSans"

`font_ps_light` == "IBMPlexSans-Light"

Usage

```
font_ps

font_ps_light
```

Format

length 1 character vector

An object of class `character` of length 1.

Note

`font_ps_light` (a.k.a. "IBMPlexSans-Light") is not available on Windows and will throw a warning if used in plots.

font_pub *Public Sans font name R variable aliases*

Description

`font_pub` == "Public Sans"

`font_pub_bold` == "Public Sans Bold"

`font_pub_light` == "Public Sans Light"

`font_pub_thin` == "Public Sans Thin"

Usage

```
font_pub

font_pub_bold

font_pub_light

font_pub_thin
```

Format

length 1 character vector

An object of class `character` of length 1.

An object of class `character` of length 1.

An object of class `character` of length 1.

Note

`font_pub_bold` (a.k.a. "`Public Sans Bold`") is not available on Windows and will throw a warning if used in plots.

`font_rc` *Roboto Condensed font name R variable aliases*

Description

`font_rc` == "`Roboto Condensed`"

`font_fc_light` == "`Roboto Condensed Light`"

Usage

```
font_rc

font_rc_light
```

Format

length 1 character vector

An object of class `character` of length 1.

Note

`font_rc_light` (a.k.a. "`Roboto Condensed Light`") is not available on Windows and will throw a warning if used in plots.

ft_cols *FT color palette*

Description

FT color palette

Usage

```
ft_cols

ft_text_col
```

Format

An object of class `list` of length 9.

An object of class `character` of length 1.

Note

don't forget you can use `scales::alpha()` with these colors

ft_geom_defaults *Change geom defaults from black to custom lights for the FT theme*

Description

Change geom defaults from black to custom lights for the FT theme

Usage

```
ft_geom_defaults()
```

ft_pal *A bright qualitative color palette*

Description

A bright qualitative color palette

Usage

```
ft_pal()
```

Examples

```
library(scales)
scales::show_col(ft_pal()(8))
```

gg_check *Spell check ggplot2 plot labels*

Description

Due to the way ggplot2 objects are created, this has to be used in a standalone context.

Usage

```
gg_check(gg, dict, ignore)
```

Arguments

gg	ggplot2 object
dict	a dictionary object or string which can be passed to hunspell::dictionary. Defaults to `hunspell::dictionary("en_US")`
ignore	character vector with additional approved words added to the dictionary. Defaults to `hunspell::en_stats`

Details

Current functionality only looks for misspelled words in the labels of ggplot2 objects. When misspelled words are found, a message is printed with the words and the label that they are in. No messages will be printed if there are no misspelled words.

Value

the object that was passed in

Examples

```
library(ggplot2)

df <- data.frame(x=c(20, 25, 30), y=c(4, 4, 4), txt=c("One", "Two", "Three"))

# not piping
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="This is some txt", y="This is more text",
       title="Thisy is a titlle",
       subtitle="This is a subtitley",
       caption="This is a captien") -> gg

gg_check(gg)
```

---

hrbrthemes-exports *hrbrthemes exported operators*

Description

The following functions are imported and then re-exported from the hrbrthemes package to enable use of the magrittr pipe operator with no additional library calls

---

import_econ_sans *Import Roboto Condensed font for use in charts*

Description

Roboto Condensed is a trademark of Google.

Usage

```
import_econ_sans()
```

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

import_goldman_sans *Import Goldman Sans font for use in charts*

Description

Goldman Sans is a trademark of Goldman Sachs and distributed under the Goldman Sachs Restricted Font License

Usage

```
import_goldman_sans()
```

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

import_inter *Import Inter font for use in charts*

Description

Inter is Copyright (c) 2016-2024 The Inter Project Authors

Usage

```
import_inter()
```

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

import_plex_sans *Import IBM Plex Sans font for use in charts*

Description

IBM Plex Sans is a trademark of IBM and distributed under the SIL Open Font License, Version 1.1.

Usage

```
import_plex_sans()
```

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

import_public_sans *Import Public Sans font for use in charts*

Description

Public Sans is Copyright 2015 Impallari Type and licensed under the SIL Open Font License, Version 1.1

Usage

```
import_public_sans()
```

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

import_roboto_condensed *Import Roboto Condensed font for use in charts*

Description

Roboto Condensed is a trademark of Google.

Usage

```
import_roboto_condensed()
```

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Note

This will take care of ensuring PDF/PostScript usage. The location of the font directory is displayed after the base import is complete. It is highly recommended that you install them on your system the same way you would any other font you wish to use in other programs.

ipsum_pal *A muted, qualitative color palette*

Description

A muted, qualitative color palette

Usage

```
ipsum_pal()
```

Examples

```
library(scales)
scales::show_col(ipsum_pal()(9))
```

modern_geom_defaults *Change geom defaults from black to white for the modern theme*

Description

Change geom defaults from black to white for the modern theme

Usage

```
modern_geom_defaults()
```

scale_color_flexoki_continuous *Continuous Color Scale Using Flexoki Colors*

Description

Continuous Color Scale Using Flexoki Colors

Usage

```
scale_color_flexoki_continuous(palette = "red", ...)
```

Arguments

palette	Name of the color palette to use ("red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")
...	Additional arguments passed to scale_color_gradientn()

Value

A continuous ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_color_flexoki_continuous(palette = "blue")
```

scale_color_flexoki_dark

Discrete Color Scale Using Flexoki Dark Colors

Description

Discrete Color Scale Using Flexoki Dark Colors

Usage

```
scale_color_flexoki_dark(...)
```

Arguments

...	Additional arguments passed to scale_color_manual()

Value

A discrete ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg, color = factor(cyl))) +
  geom_point() +
  scale_color_flexoki_dark()
```

scale_color_flexoki_dark_distiller

Distiller Color Scale Using Flexoki Dark Colors

Description

Creates a sequential color gradient based on the Flexoki dark color palette

Usage

```
scale_color_flexoki_dark_distiller(palette = "blue", direction = 1, ...)
```

Arguments

palette	Name of the color palette to use ("red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")
direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_color_gradientn()

Value

A sequential ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, color = density)) +
  geom_point() +
  scale_color_flexoki_dark_distiller(palette = "blue")
```

---

scale_color_flexoki_dark_spectrum

Distiller Color Scale Across All Flexoki Dark Colors

Description

Creates a sequential color gradient using all colors from the Flexoki dark palette

Usage

```
scale_color_flexoki_dark_spectrum(direction = 1, ...)
```

Arguments

direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_color_gradientn()

Value

A sequential ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, color = density)) +
  geom_point() +
  scale_color_flexoki_dark_spectrum()
```

scale_color_flexoki_light

Discrete Color Scale Using Flexoki Light Colors

Description

Discrete Color Scale Using Flexoki Light Colors

Usage

```
scale_color_flexoki_light(...)
```

Arguments

...	Additional arguments passed to scale_color_manual()

Value

A discrete ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg, color = factor(cyl))) +
  geom_point() +
  scale_color_flexoki_light()
```

scale_color_flexoki_light_distiller

Distiller Color Scale Using Flexoki Light Colors

Description

Creates a sequential color gradient based on the Flexoki light color palette

Usage

```
scale_color_flexoki_light_distiller(palette = "blue", direction = 1, ...)
```

Arguments

palette	Name of the color palette to use ("red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")
direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_color_gradientn()

Value

A sequential ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, color = density)) +
  geom_point() +
  scale_color_flexoki_light_distiller(palette = "blue")
```

---

scale_color_flexoki_light_spectrum

Distiller Color Scale Across All Flexoki Light Colors

Description

Creates a sequential color gradient using all colors from the Flexoki light palette

Usage

```
scale_color_flexoki_light_spectrum(direction = 1, ...)
```

Arguments

direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_color_gradientn()

Value

A sequential ggplot2 color scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, color = density)) +
  geom_point() +
  scale_color_flexoki_light_spectrum()
```

scale_colour_ft *Discrete color & fill scales based on the FT palette*

Description

See ft_pal().

Usage

```
scale_colour_ft(...)

scale_color_ft(...)

scale_fill_ft(...)
```

Arguments

...	Arguments passed on to ggplot2::discrete_scale
	aesthetics The names of the aesthetics that this scale works with.
	scale_name **[Deprecated]** The name of the scale that should be used for error messages associated with this 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::pal_hue()).
	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.
	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.
	minor_breaks One of: • NULL for no minor breaks • waiver() for the default breaks (none for discrete, one minor break between each major break for continuous) • A numeric vector of positions • A function that given the limits returns a vector of minor breaks. Also accepts rlang lambda function notation. When the function has two arguments, it will be given the limits and major break positions.
	labels One of the options below. Please note that when labels is a vector, it is highly recommended to also set the breaks argument as a vector to protect against unintended mismatches.

- 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.

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.

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.

drop Should unused factor levels be omitted from the scale? The default, TRUE, uses the levels that appear in the data; FALSE includes the levels in the factor. Please note that to display every level in a legend, the layer should use show.legend = TRUE.

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

fallback.palette Function to use when palette = NULL and the palette is not represented in the theme.

call The call used to construct the scale for reporting messages.

super The super class to use for the constructed scale

---

scale_colour_ipsum *Discrete color & fill scales based on the ipsum palette*

Description

See ipsum_pal().

Usage

```
scale_colour_ipsum(...)

scale_color_ipsum(...)

scale_fill_ipsum(...)
```

Arguments

...	Arguments passed on to ggplot2::discrete_scale
	aesthetics The names of the aesthetics that this scale works with.
	scale_name **[Deprecated]** The name of the scale that should be used for error messages associated with this 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::pal_hue()).
	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.
	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.
	minor_breaks One of: • NULL for no minor breaks • waiver() for the default breaks (none for discrete, one minor break between each major break for continuous) • A numeric vector of positions • A function that given the limits returns a vector of minor breaks. Also accepts rlang lambda function notation. When the function has two arguments, it will be given the limits and major break positions.
	labels One of the options below. Please note that when labels is a vector, it is highly recommended to also set the breaks argument as a vector to protect against unintended mismatches. • 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.
	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.
	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.

`drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` includes the levels in the factor. Please note that to display every level in a legend, the layer should use `show.legend = TRUE`.

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

`fallback.palette` Function to use when `palette = NULL` and the palette is not represented in the theme.

`call` The `call` used to construct the scale for reporting messages.

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

`scale_fill_flexoki_continuous`
Continuous Fill Scale Using Flexoki Colors

Description

Continuous Fill Scale Using Flexoki Colors

Usage

```
scale_fill_flexoki_continuous(palette = "red", ...)
```

Arguments

`palette`	Name of the color palette to use ("red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")
`...`	Additional arguments passed to scale_fill_gradientn()

Value

A continuous ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_flexoki_continuous(palette = "blue")
```

scale_fill_flexoki_dark

Discrete Fill Scale Using Flexoki Dark Colors

Description

Discrete Fill Scale Using Flexoki Dark Colors

Usage

```
scale_fill_flexoki_dark(...)
```

Arguments

...	Additional arguments passed to scale_fill_manual()

Value

A discrete ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(mtcars, aes(factor(cyl), fill = factor(cyl))) +
  geom_bar() +
  scale_fill_flexoki_dark()
```

scale_fill_flexoki_dark_distiller

Distiller Fill Scale Using Flexoki Dark Colors

Description

Creates a sequential fill gradient based on the Flexoki dark color palette

Usage

```
scale_fill_flexoki_dark_distiller(palette = "blue", direction = 1, ...)
```

Arguments

palette	Name of the color palette to use ("red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")
direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_fill_gradientn()

Value

A sequential ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_flexoki_dark_distiller(palette = "blue")
```

---

scale_fill_flexoki_dark_spectrum

Distiller Fill Scale Across All Flexoki Dark Colors

Description

Creates a sequential fill gradient using all colors from the Flexoki dark palette

Usage

```
scale_fill_flexoki_dark_spectrum(direction = 1, ...)
```

Arguments

direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_fill_gradientn()

Value

A sequential ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_flexoki_dark_spectrum()
```

scale_fill_flexoki_light

Discrete Fill Scale Using Flexoki Light Colors

Description

Discrete Fill Scale Using Flexoki Light Colors

Usage

```
scale_fill_flexoki_light(...)
```

Arguments

...	Additional arguments passed to scale_fill_manual()

Value

A discrete ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(mtcars, aes(factor(cyl), fill = factor(cyl))) +
  geom_bar() +
  scale_fill_flexoki_light()
```

scale_fill_flexoki_light_distiller

Distiller Fill Scale Using Flexoki Light Colors

Description

Creates a sequential fill gradient based on the Flexoki light color palette

Usage

```
scale_fill_flexoki_light_distiller(palette = "blue", direction = 1, ...)
```

Arguments

palette	Name of the color palette to use ("red", "orange", "yellow", "green", "cyan", "blue", "purple", "magenta")
direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_fill_gradientn()

Value

A sequential ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_flexoki_light_distiller(palette = "blue")
```

---

scale_fill_flexoki_light_spectrum

Distiller Fill Scale Across All Flexoki Light Colors

Description

Creates a sequential fill gradient using all colors from the Flexoki light palette

Usage

```
scale_fill_flexoki_light_spectrum(direction = 1, ...)
```

Arguments

direction	Sets the direction of the color scale (1 = default, -1 = reversed)
...	Additional arguments passed to scale_fill_gradientn()

Value

A sequential ggplot2 fill scale

Examples

```
library(ggplot2)
ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_tile() +
  scale_fill_flexoki_light_spectrum()
```

scale_x_percent *X & Y scales with opinionated pre-sets for percent & comma label formats*

Description

The _comma ones set comma format for axis text and expand=c(0,0) (you need to set limits).

Usage

```
scale_x_percent(
  name = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  guide = waiver(),
  n.breaks = NULL,
  labels,
  limits = NULL,
  expand = c(0.01, 0),
  oob = censor,
  na.value = NA_real_,
  trans = "identity",
  transform = "identity",
  position = "bottom",
  sec.axis = waiver(),
  accuracy = 1,
  scale = 100,
  prefix = "",
  suffix = "%",
  big.mark = " ",
  decimal.mark = ".",
  trim = TRUE,
  ...
)

scale_y_percent(
  name = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  guide = waiver(),
  n.breaks = NULL,
  labels,
  limits = NULL,
  expand = c(0.01, 0),
  oob = censor,
  na.value = NA_real_,
  trans = "identity",
```

```
  transform = "identity",
  position = "left",
  sec.axis = waiver(),
  accuracy = 1,
  scale = 100,
  prefix = "",
  suffix = "%",
  big.mark = " ",
  decimal.mark = ".",
  trim = TRUE,
  ...
)

scale_x_comma(
  name = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  guide = waiver(),
  n.breaks = NULL,
  labels,
  limits = NULL,
  expand = c(0.01, 0),
  oob = censor,
  na.value = NA_real_,
  trans = "identity",
  transform = "identity",
  position = "bottom",
  sec.axis = waiver(),
  accuracy = 1,
  scale = 1,
  prefix = "",
  suffix = "",
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  ...
)

scale_y_comma(
  name = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  guide = waiver(),
  n.breaks = NULL,
  labels,
  limits = NULL,
  expand = c(0.01, 0),
  oob = censor,
```

```
  na.value = NA_real_,
  trans = "identity",
  transform = "identity",
  position = "left",
  sec.axis = waiver(),
  accuracy = 1,
  scale = 1,
  prefix = "",
  suffix = "",
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  ...
)
```

Arguments

name	The name of the scale. Used as 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.
breaks	One of: • `NULL` for no breaks • `waiver()` for the default breaks computed by the transformation object • A numeric vector of positions • A function that takes the limits as input and returns breaks as output
minor_breaks	One of: • `NULL` for no minor breaks • `waiver()` for the default breaks (one minor break between each major break) • A numeric vector of positions • A function that given the limits returns a vector of minor breaks.
guide	guide A function used to create a guide or its name. See `ggplot2::guides()` for more information.
n.breaks	An integer guiding the number of major breaks. The algorithm may choose a slightly different number to ensure nice break labels. Will only have an effect if `breaks = waiver()`. Use NULL to use the default number of breaks given by the transformation.
labels	Specifying overrides the default format (i.e. you really don't want to do that). `NULL` means no labels.
limits	A numeric vector of length two providing limits of the scale. Use `NA` to refer to the existing minimum or maximum.
expand	same as in ggplot2
oob	Function that handles limits outside of the scale limits (out of bounds). The default replaces out of bounds values with NA.

na.value	If na.translate = TRUE, what value aesthetic value should missing be displayed as? Does not apply to position scales where NA is always placed at the far right.
trans	(DEPRECATED) Either the name of a transformation object, or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "exp", "identity", "log", "log10", "log1p", "log2", "logit", "probability", "probit", "reciprocal", "reverse" and "sqrt".
transform	Either the name of a transformation object, or the object itself. Built-in transformations include "asn", "atanh", "boxcox", "exp", "identity", "log", "log10", "log1p", "log2", "logit", "probability", "probit", "reciprocal", "reverse" and "sqrt".
position	The position of the axis. "left" or "right" for vertical scales, "top" or "bottom" for horizontal scales
sec.axis	specify a secondary axis
accuracy, scale, prefix, suffix, big.mark, decimal.mark, trim	See scales::comma_format() or scales::percent_format()
...	passed on to scales::comma_format() or scales::percent_format()

Details

The _percent ones set percent format for axis text and expand=c(0,0) (you need to set limits).

---

theme_ft_rc *A precise & pristine [ggplot2][ggplot2][ggplot2::ggplot2] theme with opinionated defaults and an emphasis on typoghraphy*

Description

You should import_roboto_condensed() first and also install the fonts on your system before trying to use this theme.

Usage

```
theme_ft_rc(
  base_family = "Roboto Condensed",
  base_size = 11.5,
  plot_title_family = base_family,
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows") "Roboto Condensed" else
    "Roboto Condensed Light",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = base_family,
```

```
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed" else
    "Roboto Condensed Light",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = base_size,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  grid = TRUE,
  axis = FALSE,
  ticks = FALSE
)

theme_modern_rc(
  base_family = "Roboto Condensed",
  base_size = 11.5,
  plot_title_family = base_family,
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows") "Roboto Condensed" else
    "Roboto Condensed Light",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = base_family,
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed" else
    "Roboto Condensed Light",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = base_size,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  grid = TRUE,
  axis = FALSE,
  ticks = FALSE
)
```

```
theme_ipsum_rc(
  base_family = "Roboto Condensed",
  base_size = 11.5,
  plot_title_family = base_family,
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows") "Roboto Condensed" else
    "Roboto Condensed Light",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = base_family,
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = if (.Platform$OS.type == "windows") "Roboto Condensed" else
    "Roboto Condensed Light",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = base_size,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  panel_spacing = grid::unit(2, "lines"),
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

`base_family`, `base_size`
: base font family and size

`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`
: plot tilte family, face, size and margin

`subtitle_family`, `subtitle_face`, `subtitle_size`
: plot subtitle family, face and size

`subtitle_margin`
: plot subtitle margin bottom (single numeric value)

`strip_text_family`, `strip_text_face`, `strip_text_size`
: facet label font family, face and size

caption_family, caption_face, caption_size, caption_margin
: plot caption family, face, size and margin

axis_text_size
: font size of axis text

axis_title_family, axis_title_face, axis_title_size
: axis title font family, face and size

axis_title_just
: axis title font justification one of [blmcrt]

plot_margin
: plot margin (specify with ggplot2::margin)

grid
: panel grid (TRUE, FALSE, or a combination of X, x, Y, y)

axis
: add x or y axes? TRUE, FALSE, "xy"

ticks
: ticks if TRUE add ticks

panel_spacing
: panel spacing (use unit())

grid_col
: grid color

axis_col
: axis color

Details

There is an option hrbrthemes.loadfonts which – if set to TRUE – will call extrafont::loadfonts() to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Why Roboto Condensed?

It's free, has tolerable kerning pairs and multiple weights. It's also different than Arial Narrow and the fonts most folks use in ggplot2 charts.

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_rc()

# seminal bar chart

# note: make this font_rc on Windows
update_geom_font_defaults(family=font_rc_light)

count(mpg, class) %>%
  ggplot(aes(class, n)) +
```

```
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_rc(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

`theme_ipsum` *A precise & pristine ggplot2 theme with opinionated defaults and an emphasis on typography*

Description

Also has a "dark" / "modern" version for the new RStudio theme

Usage

```
theme_ipsum(
  base_family = "Arial Narrow",
  base_size = 11.5,
  plot_title_family = base_family,
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
  subtitle_family = base_family,
  subtitle_size = 12,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = base_family,
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = base_family,
  caption_size = 9,
  caption_face = "italic",
  caption_margin = 10,
  axis_text_size = base_size,
  axis_title_family = subtitle_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
```

```
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

`base_family`, `base_size`
: base font family and size

`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`
: plot title family, face, size and margi

`subtitle_family`, `subtitle_face`, `subtitle_size`
: plot subtitle family, face and size

`subtitle_margin`
: plot subtitle margin bottom (single numeric value)

`strip_text_family`, `strip_text_face`, `strip_text_size`
: facet label font family, face and size

`caption_family`, `caption_face`, `caption_size`, `caption_margin`
: plot caption family, face, size and margin

`axis_text_size` font size of axis text

`axis_title_family`, `axis_title_face`, `axis_title_size`
: axis title font family, face and size

`axis_title_just`
: axis title font justification, one of `[blmcrt]`

`plot_margin` plot margin (specify with `ggplot2::margin()`)

`grid_col`, `axis_col`
: grid & axis colors; both default to `#cccccc`

`grid` panel grid (`TRUE`, `FALSE`, or a combination of `X`, `x`, `Y`, `y`)

`axis` add x or y axes? `TRUE`, `FALSE`, `"xy"`

`ticks` ticks if `TRUE` add ticks

Why Arial Narrow?

First and foremost, Arial Narrow is generally installed by default or readily available on any modern system, so it's "free"-ish; plus, it is a condensed font with solid default kerning pairs and geometric numbers.

Building upon `theme_ipsum`

The function is setup in such a way that you can customize your own one by just wrapping the call and changing the parameters. See source for examples.

Gotchas

There are distinctions between font names and various devices. Names that work for display graphics devices and bitmap ones such as png may not work well for PostScript or PDF ones. You may need two versions of a font-based theme function for them to work in a particular situation. This situation usually only arises when using a newer font with many weights but somewhat irregular internal font name patterns.

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum()

# seminal bar chart

update_geom_font_defaults()

count(mpg, class) %>%
  ggplot(aes(class, n)) +
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

---

theme_ipsum_es

A precise & pristine [ggplot2][ggplot2][ggplot2::ggplot2] theme with opinionated defaults and an emphasis on typoghraphy

---

Description

You should `import_econ_sans()` first and install the fonts on your system before trying to use this theme.

Usage

```
theme_ipsum_es(
  base_family = "EconSansCndReg",
  base_size = 11.5,
  plot_title_family = "EconSansCndBol",
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows") "EconSansCndLig" else
    "EconSansCndLig",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = base_family,
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = if (.Platform$OS.type == "windows") "EconSansCndLig" else
    "EconSansCndLig",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = base_size,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  panel_spacing = grid::unit(2, "lines"),
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

`base_family`, `base_size`
: base font family and size

`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`
: plot tilte family, face, size and margin

`subtitle_family`, `subtitle_face`, `subtitle_size`
: plot subtitle family, face and size

`subtitle_margin`
: plot subtitle margin bottom (single numeric value)

`strip_text_family`, `strip_text_face`, `strip_text_size`
: facet label font family, face and size

`caption_family`, `caption_face`, `caption_size`, `caption_margin`
: plot caption family, face, size and margin

`axis_text_size`
: font size of axis text

`axis_title_family`, `axis_title_face`, `axis_title_size`
: axis title font family, face and size

`axis_title_just`
: axis title font justification one of `[blmcrt]`

`plot_margin`
: plot margin (specify with ggplot2::margin)

`panel_spacing`
: panel spacing (use `unit()`)

`grid_col`
: grid color

`grid`
: panel grid (`TRUE`, `FALSE`, or a combination of `X`, `x`, `Y`, `y`)

`axis_col`
: axis color

`axis`
: add x or y axes? `TRUE`, `FALSE`, `"xy"`

`ticks`
: ticks if `TRUE` add ticks

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Why Econ Sans Condensed?

It's free, has tolerable kerning pairs and multiple weights. It's also different than Arial Narrow and the fonts most folks use in ggplot2 charts.

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_es()

# seminal bar chart

# note: may need to make this font_es on Windows
update_geom_font_defaults(family=font_es_light)

count(mpg, class) %>%
  ggplot(aes(class, n)) +
```

```
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_es(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

---

theme_ipsum_gs *A precise & pristine [ggplot2][ggplot2][ggplot2::ggplot2] theme with opinionated defaults and an emphasis on typoghraphy*

Description

You should `import_goldman_sans()` first and install the fonts on your system before trying to use this theme.

Usage

```
theme_ipsum_gs(
  base_family = "Goldman Sans Condensed",
  base_size = 11.5,
  plot_title_family = "Goldman Sans Condensed",
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
 subtitle_family = if (.Platform$OS.type == "windows") "Goldman Sans Condensed" else
    "Goldman Sans Condensed",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = "Goldman Sans Condensed",
  strip_text_size = 12,
  strip_text_face = "bold",
 caption_family = if (.Platform$OS.type == "windows") "Goldman Sans Condensed" else
    "Goldman Sans Condensed",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = 9,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
```

```
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

`base_family`, `base_size`	base font family and size
`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`	plot tilte family, face, size and margin
`subtitle_family`, `subtitle_face`, `subtitle_size`	plot subtitle family, face and size
`subtitle_margin`	plot subtitle margin bottom (single numeric value)
`strip_text_family`, `strip_text_face`, `strip_text_size`	facet label font family, face and size
`caption_family`, `caption_face`, `caption_size`, `caption_margin`	plot caption family, face, size and margin
`axis_text_size`	font size of axis text
`axis_title_family`, `axis_title_face`, `axis_title_size`	axis title font family, face and size
`axis_title_just`	axis title font justification one of `[blmcrt]`
`plot_margin`	plot margin (specify with ggplot2::margin)
`grid_col`	grid color
`grid`	panel grid (`TRUE`, `FALSE`, or a combination of `X`, `x`, `Y`, `y`)
`axis_col`	axis color
`axis`	add x or y axes? `TRUE`, `FALSE`, `"xy"`
`ticks`	ticks if `TRUE` add ticks

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Why Goldman Sans?

Google "goldman sans design system"

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_gs()

# seminal bar chart

update_geom_font_defaults(family=font_gs_light)

count(mpg, class) %>%
  ggplot(aes(class, n)) +
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_gs(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

`theme_ipsum_inter` *A precise & pristine [ggplot2][ggplot2][ggplot2::ggplot2] theme with opinionated defaults and an emphasis on typoghraphy*

Description

You should `import_inter()` first and install the fonts on your system before trying to use this theme.

Usage

```
theme_ipsum_inter(
  base_family = "Inter-Medium",
  base_size = 10,
  plot_title_family = "Inter-Bold",
  plot_title_size = 16,
  plot_title_face = "bold",
  plot_title_margin = 8,
```

```
  subtitle_family = "Inter-Light",
  subtitle_size = 12,
  subtitle_face = "plain",
  subtitle_margin = 13,
  strip_text_family = "Inter-SemiBold",
  strip_text_size = 12,
  strip_text_face = "bold",
  caption_family = "Inter-Thin",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_family = "Inter-Light",
  axis_text_face = "plain",
  axis_text_size = 9,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

`base_family`, `base_size`
: base font family and size

`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`
: plot tilte family, face, size and margin

`subtitle_family`, `subtitle_face`, `subtitle_size`
: plot subtitle family, face and size

`subtitle_margin`
: plot subtitle margin bottom (single numeric value)

`strip_text_family`, `strip_text_face`, `strip_text_size`
: facet label font family, face and size

`caption_family`, `caption_face`, `caption_size`, `caption_margin`
: plot caption family, face, size and margin

`axis_text_family`, `axis_text_face`
: axis text font family and face

`axis_text_size` font size of axis text

`axis_title_family`, `axis_title_face`, `axis_title_size`
: axis title font family, face and size

axis_title_just	axis title font justification one of [blmcrt]
plot_margin	plot margin (specify with ggplot2::margin)
grid_col	grid color
grid	panel grid (TRUE, FALSE, or a combination of X, x, Y, y)
axis_col	axis color
axis	add x or y axes? TRUE, FALSE, "xy"
ticks	ticks if TRUE add ticks

Details

There is an option hrbrthemes.loadfonts which – if set to TRUE – will call extrafont::loadfonts() to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Why Inter?

Google "inter font".

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_inter()

# seminal bar chart

update_geom_font_defaults(family=font_inter_medium)

count(mpg, class) %>%
  ggplot(aes(class, n)) +
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_inter(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

theme_ipsum_ps *A precise & pristine [ggplot2][ggplot2][ggplot2::ggplot2] theme with opinionated defaults and an emphasis on typoghraphy*

Description

You should `import_plex_sans()` first and install the fonts on your system before trying to use this theme.

Usage

```
theme_ipsum_ps(
  base_family = "IBMPlexSans",
  base_size = 11.5,
  plot_title_family = "IBMPlexSans-Bold",
  plot_title_size = 18,
  plot_title_face = "plain",
  plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows") "IBMPlexSans" else
    "IBMPlexSans-Light",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = "IBMPlexSans-Medium",
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = if (.Platform$OS.type == "windows") "IBMPlexSans" else
    "IBMPlexSans-Thin",
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = 9,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

base_family, base_size
: base font family and size

`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`
: plot tilte family, face, size and margin

`subtitle_family`, `subtitle_face`, `subtitle_size`
: plot subtitle family, face and size

`subtitle_margin`
: plot subtitle margin bottom (single numeric value)

`strip_text_family`, `strip_text_face`, `strip_text_size`
: facet label font family, face and size

`caption_family`, `caption_face`, `caption_size`, `caption_margin`
: plot caption family, face, size and margin

`axis_text_size`
: font size of axis text

`axis_title_family`, `axis_title_face`, `axis_title_size`
: axis title font family, face and size

`axis_title_just`
: axis title font justification one of `[blmcrt]`

`plot_margin`
: plot margin (specify with ggplot2::margin)

`grid_col`
: grid color

`grid`
: panel grid (`TRUE`, `FALSE`, or a combination of `X`, `x`, `Y`, `y`)

`axis_col`
: axis color

`axis`
: add x or y axes? `TRUE`, `FALSE`, "`xy`"

`ticks`
: ticks if `TRUE` add ticks

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Why IBM Plex Sans?

It's free, has tolerable kerning pairs and multiple weights. It's also different "not Helvetica".

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_rc()
```

```
# seminal bar chart

# note: make this font_rc on Windows
update_geom_font_defaults(family=font_rc_light)

count(mpg, class) %>%
  ggplot(aes(class, n)) +
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_rc(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

---

theme_ipsum_pub *A precise & pristine [ggplot2][ggplot2][ggplot2::ggplot2] theme with opinionated defaults and an emphasis on typoghraphy*

Description

You should `import_public_sans()` first and install the fonts on your system before trying to use this theme.

Usage

```
theme_ipsum_pub(
  base_family = "Public Sans",
  base_size = 10.5,
  plot_title_family = if (.Platform$OS.type == "windows") "Public Sans" else
    "Public Sans Bold",
  plot_title_size = 18,
  plot_title_face = "bold",
  plot_title_margin = 10,
  subtitle_family = if (.Platform$OS.type == "windows") "Public Sans Thin" else
    "Public Sans Thin",
  subtitle_size = 13,
  subtitle_face = "plain",
  subtitle_margin = 15,
  strip_text_family = base_family,
  strip_text_size = 12,
  strip_text_face = "plain",
  caption_family = if (.Platform$OS.type == "windows") "Public Sans Thin" else
    "Public Sans Thin",
```

```
  caption_size = 9,
  caption_face = "plain",
  caption_margin = 10,
  axis_text_size = base_size,
  axis_title_family = base_family,
  axis_title_size = 9,
  axis_title_face = "plain",
  axis_title_just = "rt",
  plot_margin = margin(30, 30, 30, 30),
  grid_col = "#cccccc",
  grid = TRUE,
  axis_col = "#cccccc",
  axis = FALSE,
  ticks = FALSE
)
```

Arguments

`base_family`, `base_size`
: base font family and size

`plot_title_family`, `plot_title_face`, `plot_title_size`, `plot_title_margin`
: plot tilte family, face, size and margin

`subtitle_family`, `subtitle_face`, `subtitle_size`
: plot subtitle family, face and size

`subtitle_margin`
: plot subtitle margin bottom (single numeric value)

`strip_text_family`, `strip_text_face`, `strip_text_size`
: facet label font family, face and size

`caption_family`, `caption_face`, `caption_size`, `caption_margin`
: plot caption family, face, size and margin

`axis_text_size`
: font size of axis text

`axis_title_family`, `axis_title_face`, `axis_title_size`
: axis title font family, face and size

`axis_title_just`
: axis title font justification one of `[blmcrt]`

`plot_margin`
: plot margin (specify with ggplot2::margin)

`grid_col`
: grid color

`grid`
: panel grid (`TRUE`, `FALSE`, or a combination of `X`, `x`, `Y`, `y`)

`axis_col`
: axis color

`axis`
: add x or y axes? `TRUE`, `FALSE`, `"xy"`

`ticks`
: ticks if `TRUE` add ticks

Details

There is an option `hrbrthemes.loadfonts` which – if set to `TRUE` – will call `extrafont::loadfonts()` to register non-core fonts with R PDF & PostScript devices. If you are running under Windows, the package calls the same function to register non-core fonts with the Windows graphics device.

Why Public Sans?

See the design principles.

Examples

```
## Not run:
library(ggplot2)
library(dplyr)

# seminal scatterplot
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 scatterplot example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_pub()

# seminal bar chart

update_geom_font_defaults(family=font_pub)

count(mpg, class) %>%
  ggplot(aes(class, n)) +
  geom_col() +
  geom_text(aes(label=n), nudge_y=3) +
  labs(x="Fuel efficiency (mpg)", y="Weight (tons)",
       title="Seminal ggplot2 bar chart example",
       subtitle="A plot that is only useful for demonstration purposes",
       caption="Brought to you by the letter 'g'") +
  theme_ipsum_pub(grid="Y") +
  theme(axis.text.y=element_blank())

## End(Not run)
```

`update_geom_font_defaults` *Update matching font defaults for text geoms*

Description

Updates ggplot2::geom_label and ggplot2::geom_text font defaults

Usage

```
update_geom_font_defaults(
  family = "Arial Narrow",
  face = "plain",
  size = 3.5,
  color = "#2b2b2b"
)
```

Arguments

`family`, `face`, `size`, `color`
: font family name, face, size and color

Index