

Package ‘ggrefine’

April 14, 2026

Title Pretty 'ggplot2' Themes

Version 0.2.0

Description A set of complete 'ggplot2' themes and refine functions for plot appearance. The themes are designed to work seamlessly with the refine functions, which adjust gridlines and axis elements to suit the particulars of a plot.

License MIT + file LICENSE

URL <https://github.com/davidhodge931/ggrefine>,
<https://davidhodge931.github.io/ggrefine/>

BugReports <https://github.com/davidhodge931/ggrefine/issues>

Depends R (>= 4.1.0)

Imports blends, flexoki, ggplot2, grid, jumble, rlang, scales, viridis

Suggests patchwork

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

NeedsCompilation no

Author David Hodge [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-3868-7501>>)

Maintainer David Hodge <davidhodge931@gmail.com>

Repository CRAN

Date/Publication 2026-04-14 09:20:02 UTC

Contents

refine_classic	2
refine_fusion	3
refine_modern	4
refine_none	6

refine_void	7
theme_dark	8
theme_grey	11
theme_light	14

Index 18

refine_classic	<i>classic refine</i>
----------------	-----------------------

Description

Removes gridlines and ticks from discrete axes.

Usage

```
refine_classic(x_type = "continuous", y_type = "continuous", focus = NULL, ...)
```

Arguments

<code>x_type</code>	Character. Type of x-axis: "continuous", "binned", or "discrete".
<code>y_type</code>	Character. Type of y-axis: "continuous", "binned", or "discrete".
<code>focus</code>	Character. The primary axis of interest: "x" or "y". Gridlines and axis elements are removed from the opposite axis. If NULL (default), focus is inferred from <code>x_type</code> and <code>y_type</code> : discrete x with continuous/binned y gives "x", continuous/binned x with discrete y gives "y", otherwise "x".
<code>...</code>	Additional arguments (currently unused).

Value

A ggplot2 theme object

Examples

```
library(ggplot2)

set_theme(new = theme_grey())

p_continuous <- mpg |>
  ggplot(aes(x = displ, y = hwy)) +
  geom_point(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_x <- mpg |>
  ggplot(aes(x = drv, y = hwy)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_y <- mpg |>
  ggplot(aes(x = hwy, y = drv)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))
```

```

patchwork::wrap_plots(
  p_continuous + ggrefine::refine_modern() + labs(title = "ggrefine::refine_modern"),
  p_discrete_x + ggrefine::refine_modern(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_modern(y_type = "discrete"),
  p_continuous + ggrefine::refine_classic() + labs(title = "ggrefine::refine_classic"),
  p_discrete_x + ggrefine::refine_classic(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_classic(y_type = "discrete"),
  p_continuous + ggrefine::refine_fusion() + labs(title = "ggrefine::refine_fusion"),
  p_discrete_x + ggrefine::refine_fusion(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_fusion(y_type = "discrete"),
  p_continuous + ggrefine::refine_void() + labs(title = "ggrefine::refine_void"),
  p_discrete_x + ggrefine::refine_void(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_void(y_type = "discrete"),
  p_continuous + ggrefine::refine_none() + labs(title = "ggrefine::refine_none"),
  p_discrete_x + ggrefine::refine_none(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_none(y_type = "discrete"),
  ncol = 3
)

```

refine_fusion

Fusion refine

Description

Similar to [refine_modern\(\)](#), but keeps gridlines on both axes when both are continuous or binned. A fusion between [refine_modern\(\)](#) and [refine_classic\(\)](#).

Usage

```
refine_fusion(x_type = "continuous", y_type = "continuous", focus = NULL, ...)
```

Arguments

<code>x_type</code>	Character. Type of x-axis: "continuous", "binned", or "discrete".
<code>y_type</code>	Character. Type of y-axis: "continuous", "binned", or "discrete".
<code>focus</code>	Character. The primary axis of interest: "x" or "y". Gridlines and axis elements are removed from the opposite axis. If NULL (default), focus is inferred from <code>x_type</code> and <code>y_type</code> : discrete x with continuous/binned y gives "x", continuous/binned x with discrete y gives "y", otherwise "x".
<code>...</code>	Additional arguments (currently unused).

Value

A ggplot2 theme object

Examples

```

library(ggplot2)

set_theme(new = theme_grey())

p_continuous <- mpg |>
  ggplot(aes(x = displ, y = hwy)) +
  geom_point(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_x <- mpg |>
  ggplot(aes(x = drv, y = hwy)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_y <- mpg |>
  ggplot(aes(x = hwy, y = drv)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

patchwork::wrap_plots(
  p_continuous + ggrefine::refine_modern() + labs(title = "ggrefine::refine_modern"),
  p_discrete_x + ggrefine::refine_modern(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_modern(y_type = "discrete"),
  p_continuous + ggrefine::refine_classic() + labs(title = "ggrefine::refine_classic"),
  p_discrete_x + ggrefine::refine_classic(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_classic(y_type = "discrete"),
  p_continuous + ggrefine::refine_fusion() + labs(title = "ggrefine::refine_fusion"),
  p_discrete_x + ggrefine::refine_fusion(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_fusion(y_type = "discrete"),
  p_continuous + ggrefine::refine_void() + labs(title = "ggrefine::refine_void"),
  p_discrete_x + ggrefine::refine_void(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_void(y_type = "discrete"),
  p_continuous + ggrefine::refine_none() + labs(title = "ggrefine::refine_none"),
  p_discrete_x + ggrefine::refine_none(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_none(y_type = "discrete"),
  ncol = 3
)

```

refine_modern

Modern refine

Description

Removes gridlines and axis line/tick elements from the non-focused dimension. Also removes ticks on discrete axes.

Usage

```
refine_modern(x_type = "continuous", y_type = "continuous", focus = NULL, ...)
```

Arguments

<code>x_type</code>	Character. Type of x-axis: "continuous", "binned", or "discrete".
<code>y_type</code>	Character. Type of y-axis: "continuous", "binned", or "discrete".
<code>focus</code>	Character. The primary axis of interest: "x" or "y". Gridlines and axis elements are removed from the opposite axis. If NULL (default), focus is inferred from <code>x_type</code> and <code>y_type</code> : discrete x with continuous/binned y gives "x", continuous/binned x with discrete y gives "y", otherwise "x".
<code>...</code>	Additional arguments (currently unused).

Value

A `ggplot2` theme object

Examples

```
library(ggplot2)

set_theme(new = theme_grey())

p_continuous <- mpg |>
  ggplot(aes(x = displ, y = hwy)) +
  geom_point(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_x <- mpg |>
  ggplot(aes(x = drv, y = hwy)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_y <- mpg |>
  ggplot(aes(x = hwy, y = drv)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

patchwork::wrap_plots(
  p_continuous + ggrefine::refine_modern() + labs(title = "ggrefine::refine_modern"),
  p_discrete_x + ggrefine::refine_modern(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_modern(y_type = "discrete"),
  p_continuous + ggrefine::refine_classic() + labs(title = "ggrefine::refine_classic"),
  p_discrete_x + ggrefine::refine_classic(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_classic(y_type = "discrete"),
  p_continuous + ggrefine::refine_fusion() + labs(title = "ggrefine::refine_fusion"),
  p_discrete_x + ggrefine::refine_fusion(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_fusion(y_type = "discrete"),
  p_continuous + ggrefine::refine_void() + labs(title = "ggrefine::refine_void"),
  p_discrete_x + ggrefine::refine_void(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_void(y_type = "discrete"),
  p_continuous + ggrefine::refine_none() + labs(title = "ggrefine::refine_none"),
  p_discrete_x + ggrefine::refine_none(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_none(y_type = "discrete"),
  ncol = 3
)
```

refine_none	<i>No refine</i>
-------------	------------------

Description

Leaves the theme unchanged.

Usage

```
refine_none(x_type = "continuous", y_type = "continuous", focus = NULL, ...)
```

Arguments

x_type	Character. Type of x-axis: "continuous", "binned", or "discrete".
y_type	Character. Type of y-axis: "continuous", "binned", or "discrete".
focus	Character. The primary axis of interest: "x" or "y". Gridlines and axis elements are removed from the opposite axis. If NULL (default), focus is inferred from x_type and y_type: discrete x with continuous/binned y gives "x", continuous/binned x with discrete y gives "y", otherwise "x".
...	Additional arguments (currently unused).

Value

An empty ggplot2 theme object

Examples

```
library(ggplot2)

set_theme(new = theme_grey())

p_continuous <- mpg |>
  ggplot(aes(x = displ, y = hwy)) +
  geom_point(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_x <- mpg |>
  ggplot(aes(x = drv, y = hwy)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_y <- mpg |>
  ggplot(aes(x = hwy, y = drv)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

patchwork::wrap_plots(
  p_continuous + ggrefine::refine_modern() + labs(title = "ggrefine::refine_modern"),
  p_discrete_x + ggrefine::refine_modern(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_modern(y_type = "discrete"),
  p_continuous + ggrefine::refine_classic() + labs(title = "ggrefine::refine_classic"),
```

```

p_discrete_x + ggrefine::refine_classic(x_type = "discrete"),
p_discrete_y + ggrefine::refine_classic(y_type = "discrete"),
p_continuous + ggrefine::refine_fusion() + labs(title = "ggrefine::refine_fusion"),
p_discrete_x + ggrefine::refine_fusion(x_type = "discrete"),
p_discrete_y + ggrefine::refine_fusion(y_type = "discrete"),
p_continuous + ggrefine::refine_void() + labs(title = "ggrefine::refine_void"),
p_discrete_x + ggrefine::refine_void(x_type = "discrete"),
p_discrete_y + ggrefine::refine_void(y_type = "discrete"),
p_continuous + ggrefine::refine_none() + labs(title = "ggrefine::refine_none"),
p_discrete_x + ggrefine::refine_none(x_type = "discrete"),
p_discrete_y + ggrefine::refine_none(y_type = "discrete"),
ncol = 3
)

```

refine_void

Void refine

Description

Removes axes and gridlines.

Usage

```
refine_void(x_type = "continuous", y_type = "continuous", focus = NULL, ...)
```

Arguments

<code>x_type</code>	Character. Type of x-axis: "continuous", "binned", or "discrete".
<code>y_type</code>	Character. Type of y-axis: "continuous", "binned", or "discrete".
<code>focus</code>	Character. The primary axis of interest: "x" or "y". Gridlines and axis elements are removed from the opposite axis. If NULL (default), focus is inferred from <code>x_type</code> and <code>y_type</code> : discrete x with continuous/binned y gives "x", continuous/binned x with discrete y gives "y", otherwise "x".
<code>...</code>	Additional arguments (currently unused).

Value

A ggplot2 theme object

Examples

```

library(ggplot2)

set_theme(new = theme_grey())

p_continuous <- mpg |>
  ggplot(aes(x = displ, y = hwy)) +

```

```

geom_point(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_x <- mpg |>
  ggplot(aes(x = drv, y = hwy)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

p_discrete_y <- mpg |>
  ggplot(aes(x = hwy, y = drv)) +
  geom_jitter(shape = 21, colour = blends::multiply("#357BA2FF"))

patchwork::wrap_plots(
  p_continuous + ggrefine::refine_modern() + labs(title = "ggrefine::refine_modern"),
  p_discrete_x + ggrefine::refine_modern(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_modern(y_type = "discrete"),
  p_continuous + ggrefine::refine_classic() + labs(title = "ggrefine::refine_classic"),
  p_discrete_x + ggrefine::refine_classic(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_classic(y_type = "discrete"),
  p_continuous + ggrefine::refine_fusion() + labs(title = "ggrefine::refine_fusion"),
  p_discrete_x + ggrefine::refine_fusion(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_fusion(y_type = "discrete"),
  p_continuous + ggrefine::refine_void() + labs(title = "ggrefine::refine_void"),
  p_discrete_x + ggrefine::refine_void(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_void(y_type = "discrete"),
  p_continuous + ggrefine::refine_none() + labs(title = "ggrefine::refine_none"),
  p_discrete_x + ggrefine::refine_none(x_type = "discrete"),
  p_discrete_y + ggrefine::refine_none(y_type = "discrete"),
  ncol = 3
)

```

 theme_dark

Darker theme

Description

A complete theme for a dark panel background.

Usage

```

theme_dark(
  ...,
  text_size = 10,
  text_family = "",
  text_colour = flexoki::flexoki$base["base200"],
  legend_place = "right",
  legend_axis_line_colour = plot_background_fill,
  legend_axis_line_linewidth = axis_line_linewidth,
  legend_background_fill = plot_background_fill,
  legend_key_fill = plot_background_fill,

```

```

legend_ticks_colour = legend_axis_line_colour,
legend_ticks_linewidth = legend_axis_line_linewidth,
legend_ticks_length = grid::unit(c(2.75, 0), "pt"),
axis_line_colour = flexoki::flexoki$base["base600"],
axis_line_linewidth = 0.25,
axis_ticks_colour = axis_line_colour,
axis_ticks_linewidth = axis_line_linewidth,
axis_ticks_length = grid::unit(3.66, "pt"),
panel_background_fill = flexoki::flexoki$base["base950"],
panel_grid_colour = "black",
panel_grid_linetype = 1,
panel_grid_linewidth = 1,
panel_grid_minor_linetype = 1,
panel_grid_minor_linewidth = 0.5,
plot_background_fill = "black",
geom_fill = "#357BA2FF",
geom_colour = geom_fill,
palette_fill_discrete = jumble::jumble,
palette_colour_discrete = palette_fill_discrete,
palette_fill_continuous = viridis::turbo(n = 256),
palette_colour_continuous = palette_fill_continuous,
panel_widths = NULL,
panel_heights = NULL
)

```

Arguments

... Require named arguments (and support trailing commas).

text_size The base size of the text theme element. Defaults to 10.

text_family The base family of the text theme element. Defaults to "".

text_colour The base colour of the text theme element.

legend_place The place of the legend. Either "right", "top" or "bottom".

legend_axis_line_colour
The colour of the legend.axis.line theme element.

legend_axis_line_linewidth
The linewidth of the legend.axis.line theme element.

legend_background_fill
The fill (and colour) of the legend.background theme element.

legend_key_fill
The fill (and colour) of the legend.key theme element.

legend_ticks_colour
The colour of the legend.ticks theme element.

legend_ticks_linewidth
The linewidth of the legend.ticks theme element.

legend_ticks_length
The legend.ticks.length theme element.

axis_line_colour	The colour of the axis.line theme element.
axis_line_linewidth	The linewidth of the axis.line theme element.
axis_ticks_colour	The colour of the axis.ticks theme element.
axis_ticks_linewidth	The linewidth of the axis.ticks theme element.
axis_ticks_length	The length of the axis.ticks.length theme element.
panel_background_fill	The fill (and colour) of the panel.background theme element.
panel_grid_colour	The colour of the panel.grid theme element.
panel_grid_linetype	The linetype of the panel.grid.major theme element.
panel_grid_linewidth	The linewidth of the panel.grid.major theme element.
panel_grid_minor_linetype	The linetype of the panel.grid.minor theme element.
panel_grid_minor_linewidth	The linewidth of the panel.grid.minor theme element.
plot_background_fill	The fill (and colour) of the plot.background theme element.
geom_fill	The default fill colour of geom elements.
geom_colour	The default border colour of geom elements. Defaults to geom_fill.
palette_fill_discrete	The default discrete fill palette. A function or vector of colours.
palette_colour_discrete	The default discrete colour palette. Defaults to palette_fill_discrete.
palette_fill_continuous	The default continuous fill palette. A vector of colours.
palette_colour_continuous	The default continuous colour palette. Defaults to palette_fill_continuous.
panel_widths	The panel.widths theme element. A unit or unit vector setting the width of individual panels, or a single unit for the total panel area width. Overrides aspect ratio set by the theme, coord, or facets. Defaults to NULL.
panel_heights	The panel.heights theme element. A unit or unit vector setting the height of individual panels, or a single unit for the total panel area height. Overrides aspect ratio set by the theme, coord, or facets. Defaults to NULL.

Value

A ggplot theme.

Examples

```

library(ggplot2)

p_base_light <- mpg |>
  ggplot(aes(x = hwy)) +
  geom_histogram(
    stat = "bin", shape = 21,
    colour = blends::multiply("#357BA2FF")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

p_base_dark <- mpg |>
  ggplot(aes(x = hwy)) +
  geom_histogram(
    stat = "bin", shape = 21,
    colour = blends::screen("#357BA2FF")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

p_light <- p_base_light + ggrefine::theme_light() + labs(title = "ggrefine::theme_light")
p_dark <- p_base_dark + ggrefine::theme_dark() + labs(title = "ggrefine::theme_dark")
p_grey <- p_base_light + ggrefine::theme_grey() + labs(title = "ggrefine::theme_grey")

patchwork::wrap_plots(
  p_light,
  p_dark,
  p_grey
)

```

 theme_grey

Greyer theme

Description

A complete theme with a grey panel background on a white plot background. The panel grid colour is derived automatically by blending `panel_background_fill` with itself using `blends::multiply()`, producing a subtly darker tone that stays harmonious with the panel colour.

Usage

```

theme_grey(
  ...,
  text_size = 10,
  text_family = "",
  text_colour = flexoki::flexoki$base["black"],
  legend_place = "right",
  legend_axis_line_colour = plot_background_fill,
  legend_axis_line_linewidth = axis_line_linewidth,

```

```

legend_background_fill = plot_background_fill,
legend_key_fill = plot_background_fill,
legend_ticks_colour = legend_axis_line_colour,
legend_ticks_linewidth = legend_axis_line_linewidth,
legend_ticks_length = grid::unit(c(2.75, 0), "pt"),
axis_line_colour = flexoki::flexoki$base["base600"],
axis_line_linewidth = 0.25,
axis_ticks_colour = axis_line_colour,
axis_ticks_linewidth = axis_line_linewidth,
axis_ticks_length = grid::unit(3.66, "pt"),
panel_background_fill = "#EAEFFFFF",
panel_grid_colour = blends::multiply(panel_background_fill),
panel_grid_linetype = 1,
panel_grid_linewidth = 1,
panel_grid_minor_linetype = 1,
panel_grid_minor_linewidth = 0.5,
plot_background_fill = "white",
geom_fill = "#357BA2FF",
geom_colour = geom_fill,
palette_fill_discrete = jumble::jumble,
palette_colour_discrete = palette_fill_discrete,
palette_fill_continuous = viridis::turbo(n = 256),
palette_colour_continuous = palette_fill_continuous,
panel_widths = NULL,
panel_heights = NULL
)

```

Arguments

...	Require named arguments (and support trailing commas).
text_size	The base size of the text theme element. Defaults to 10.
text_family	The base family of the text theme element. Defaults to "".
text_colour	The base colour of the text theme element.
legend_place	The place of the legend. Either "right", "top" or "bottom".
legend_axis_line_colour	The colour of the legend.axis.line theme element.
legend_axis_line_linewidth	The linewidth of the legend.axis.line theme element.
legend_background_fill	The fill (and colour) of the legend.background theme element.
legend_key_fill	The fill (and colour) of the legend.key theme element.
legend_ticks_colour	The colour of the legend.ticks theme element.
legend_ticks_linewidth	The linewidth of the legend.ticks theme element.

legend_ticks_length	The legend.ticks.length theme element.
axis_line_colour	The colour of the axis.line theme element.
axis_line_linewidth	The linewidth of the axis.line theme element.
axis_ticks_colour	The colour of the axis.ticks theme element.
axis_ticks_linewidth	The linewidth of the axis.ticks theme element.
axis_ticks_length	The length of the axis.ticks.length theme element.
panel_background_fill	The fill (and colour) of the panel.background theme element.
panel_grid_colour	The colour of the panel.grid theme element.
panel_grid_linetype	The linetype of the panel.grid.major theme element.
panel_grid_linewidth	The linewidth of the panel.grid.major theme element.
panel_grid_minor_linetype	The linetype of the panel.grid.minor theme element.
panel_grid_minor_linewidth	The linewidth of the panel.grid.minor theme element.
plot_background_fill	The fill (and colour) of the plot.background theme element.
geom_fill	The default fill colour of geom elements.
geom_colour	The default border colour of geom elements. Defaults to geom_fill.
palette_fill_discrete	The default discrete fill palette. A function or vector of colours.
palette_colour_discrete	The default discrete colour palette. Defaults to palette_fill_discrete.
palette_fill_continuous	The default continuous fill palette. A vector of colours.
palette_colour_continuous	The default continuous colour palette. Defaults to palette_fill_continuous.
panel_widths	The panel.widths theme element. A unit or unit vector setting the width of individual panels, or a single unit for the total panel area width. Overrides aspect ratio set by the theme, coord, or facets. Defaults to NULL.
panel_heights	The panel.heights theme element. A unit or unit vector setting the height of individual panels, or a single unit for the total panel area height. Overrides aspect ratio set by the theme, coord, or facets. Defaults to NULL.

Value

A ggplot theme.

Examples

```

library(ggplot2)

p_base_light <- mpg |>
  ggplot(aes(x = hwy)) +
  geom_histogram(
    stat = "bin", shape = 21,
    colour = blends::multiply("#357BA2FF")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

p_base_dark <- mpg |>
  ggplot(aes(x = hwy)) +
  geom_histogram(
    stat = "bin", shape = 21,
    colour = blends::screen("#357BA2FF")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

p_light <- p_base_light + ggrefine::theme_light() + labs(title = "ggrefine::theme_light")
p_dark <- p_base_dark + ggrefine::theme_dark() + labs(title = "ggrefine::theme_dark")
p_grey <- p_base_light + ggrefine::theme_grey() + labs(title = "ggrefine::theme_grey")

patchwork::wrap_plots(
  p_light,
  p_dark,
  p_grey
)

```

 theme_light

Lighter theme

Description

A complete theme for a white panel background.

Usage

```

theme_light(
  ...,
  text_size = 10,
  text_family = "",
  text_colour = flexoki::flexoki$base["black"],
  legend_place = "right",
  legend_axis_line_colour = NULL,
  legend_axis_line_linewidth = NULL,
  legend_background_fill = NULL,
  legend_key_fill = NULL,

```

```

legend_ticks_colour = NULL,
legend_ticks_linewidth = NULL,
legend_ticks_length = grid::unit(c(2.75, 0), "pt"),
axis_line_colour = flexoki::flexoki$base["base600"],
axis_line_linewidth = 0.25,
axis_ticks_colour = NULL,
axis_ticks_linewidth = NULL,
axis_ticks_length = grid::unit(3.66, "pt"),
panel_background_fill = "white",
panel_grid_colour = flexoki::flexoki$base["base50"],
panel_grid_linetype = 1,
panel_grid_linewidth = 1,
panel_grid_minor_linetype = 1,
panel_grid_minor_linewidth = 0.5,
plot_background_fill = "white",
geom_fill = "#357BA2FF",
geom_colour = geom_fill,
palette_fill_discrete = jumble::jumble,
palette_colour_discrete = palette_fill_discrete,
palette_fill_continuous = viridis::turbo(n = 256),
palette_colour_continuous = palette_fill_continuous,
panel_widths = NULL,
panel_heights = NULL
)

```

Arguments

... Require named arguments (and support trailing commas).

text_size The base size of the text theme element. Defaults to 10.

text_family The base family of the text theme element. Defaults to "".

text_colour The base colour of the text theme element.

legend_place The place of the legend. Either "right", "top" or "bottom".

legend_axis_line_colour
The colour of the legend.axis.line theme element.

legend_axis_line_linewidth
The linewidth of the legend.axis.line theme element.

legend_background_fill
The fill (and colour) of the legend.background theme element.

legend_key_fill
The fill (and colour) of the legend.key theme element.

legend_ticks_colour
The colour of the legend.ticks theme element.

legend_ticks_linewidth
The linewidth of the legend.ticks theme element.

legend_ticks_length
The legend.ticks.length theme element.

axis_line_colour	The colour of the axis.line theme element.
axis_line_linewidth	The linewidth of the axis.line theme element.
axis_ticks_colour	The colour of the axis.ticks theme element.
axis_ticks_linewidth	The linewidth of the axis.ticks theme element.
axis_ticks_length	The length of the axis.ticks.length theme element.
panel_background_fill	The fill (and colour) of the panel.background theme element.
panel_grid_colour	The colour of the panel.grid theme element.
panel_grid_linetype	The linetype of the panel.grid.major theme element.
panel_grid_linewidth	The linewidth of the panel.grid.major theme element.
panel_grid_minor_linetype	The linetype of the panel.grid.minor theme element.
panel_grid_minor_linewidth	The linewidth of the panel.grid.minor theme element.
plot_background_fill	The fill (and colour) of the plot.background theme element.
geom_fill	The default fill colour of geom elements.
geom_colour	The default border colour of geom elements. Defaults to geom_fill.
palette_fill_discrete	The default discrete fill palette. A function or vector of colours.
palette_colour_discrete	The default discrete colour palette. Defaults to palette_fill_discrete.
palette_fill_continuous	The default continuous fill palette. A vector of colours.
palette_colour_continuous	The default continuous colour palette. Defaults to palette_fill_continuous.
panel_widths	The panel.widths theme element. A unit or unit vector setting the width of individual panels, or a single unit for the total panel area width. Overrides aspect ratio set by the theme, coord, or facets. Defaults to NULL.
panel_heights	The panel.heights theme element. A unit or unit vector setting the height of individual panels, or a single unit for the total panel area height. Overrides aspect ratio set by the theme, coord, or facets. Defaults to NULL.

Value

A ggplot theme.

Examples

```
library(ggplot2)

p_base_light <- mpg |>
  ggplot(aes(x = hwy)) +
  geom_histogram(
    stat = "bin", shape = 21,
    colour = blends::multiply("#357BA2FF")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

p_base_dark <- mpg |>
  ggplot(aes(x = hwy)) +
  geom_histogram(
    stat = "bin", shape = 21,
    colour = blends::screen("#357BA2FF")
  ) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05)))

p_light <- p_base_light + ggrefine::theme_light() + labs(title = "ggrefine::theme_light")
p_dark <- p_base_dark + ggrefine::theme_dark() + labs(title = "ggrefine::theme_dark")
p_grey <- p_base_light + ggrefine::theme_grey() + labs(title = "ggrefine::theme_grey")

patchwork::wrap_plots(
  p_light,
  p_dark,
  p_grey
)
```

Index

`refine_classic`, 2
`refine_classic()`, 3
`refine_fusion`, 3
`refine_modern`, 4
`refine_modern()`, 3
`refine_none`, 6
`refine_void`, 7

`theme_dark`, 8
`theme_grey`, 11
`theme_light`, 14