Package 'ggrefine'

March 30, 2026

Title Pretty 'ggplot2' Themes

Version 0.1.0

Description A set of complete 'ggplot2' themes and functions to refine these.

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 2026-03-30 18:00:02 UTC

Contents

`refine_classic` *classic refine*

Description

Removes gridlines and ticks from discrete axes.

Usage

```
refine_classic(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

A ggplot2 theme object

Examples

```
library(ggplot2)

set_theme(new = theme_stone())

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 + refine_modern() + labs(title = "refine_modern"),
  p_discrete_x + refine_modern(x_type = "discrete"),
  p_discrete_y + refine_modern(y_type = "discrete"),
```

```
  p_continuous + refine_classic() + labs(title = "refine_classic"),
  p_discrete_x + refine_classic(x_type = "discrete"),
  p_discrete_y + refine_classic(y_type = "discrete"),
  p_continuous + refine_fusion() + labs(title = "refine_fusion"),
  p_discrete_x + refine_fusion(x_type = "discrete"),
  p_discrete_y + refine_fusion(y_type = "discrete"),
  p_continuous + refine_void() + labs(title = "refine_void"),
  p_discrete_x + refine_void(x_type = "discrete"),
  p_discrete_y + refine_void(y_type = "discrete"),
  p_continuous + refine_none() + labs(title = "refine_none"),
  p_discrete_x + refine_none(x_type = "discrete"),
  p_discrete_y + 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

`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

A ggplot2 theme object

Examples

```
library(ggplot2)

set_theme(new = theme_stone())

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 + refine_modern() + labs(title = "refine_modern"),
  p_discrete_x + refine_modern(x_type = "discrete"),
  p_discrete_y + refine_modern(y_type = "discrete"),
  p_continuous + refine_classic() + labs(title = "refine_classic"),
  p_discrete_x + refine_classic(x_type = "discrete"),
  p_discrete_y + refine_classic(y_type = "discrete"),
  p_continuous + refine_fusion() + labs(title = "refine_fusion"),
  p_discrete_x + refine_fusion(x_type = "discrete"),
  p_discrete_y + refine_fusion(y_type = "discrete"),
  p_continuous + refine_void() + labs(title = "refine_void"),
  p_discrete_x + refine_void(x_type = "discrete"),
  p_discrete_y + refine_void(y_type = "discrete"),
  p_continuous + refine_none() + labs(title = "refine_none"),
  p_discrete_x + refine_none(x_type = "discrete"),
  p_discrete_y + 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

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

A ggplot2 theme object

Examples

```
library(ggplot2)

set_theme(new = theme_stone())

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

refine_none *No refine*

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_stone())

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

```
  p_discrete_x + refine_classic(x_type = "discrete"),
  p_discrete_y + refine_classic(y_type = "discrete"),
  p_continuous + refine_fusion() + labs(title = "refine_fusion"),
  p_discrete_x + refine_fusion(x_type = "discrete"),
  p_discrete_y + refine_fusion(y_type = "discrete"),
  p_continuous + refine_void() + labs(title = "refine_void"),
  p_discrete_x + refine_void(x_type = "discrete"),
  p_discrete_y + refine_void(y_type = "discrete"),
  p_continuous + refine_none() + labs(title = "refine_none"),
  p_discrete_x + refine_none(x_type = "discrete"),
  p_discrete_y + 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

`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

A ggplot2 theme object

Examples

```
library(ggplot2)

set_theme(new = theme_stone())

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

`theme_black` *Black theme*

Description

A complete theme for a dark panel background.

Usage

```
theme_black(
  ...,
  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)
library(ggrefine)

p_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_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_white  <- p_light + theme_white() + labs(title = "theme_white")
p_oat    <- p_light + theme_oat()   + labs(title = "theme_oat")
p_stone <- p_light + theme_stone() + labs(title = 'theme_stone')
p_black  <- p_dark  + theme_black() + labs(title = "theme_black")

patchwork::wrap_plots(
  p_white,
  p_black,
  p_oat,
  p_stone
)
```

---

theme_oat *Oat theme*

Description

A complete theme with a tinted panel 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. Pass any colour to `panel_background_fill` to change the tint — the grid will adjust accordingly.

Usage

```
theme_oat(
  ...,
  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 = flexoki::flexoki$base["base50"],
  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)

p1 <- penguins |>
  ggplot(aes(x = species, y = body_mass, colour = species, fill = species)) +
  geom_jitter(shape = 21) +
  scale_colour_discrete(palette = blends::multiply(scales::pal_hue()))

# Default: flexoki base50 oat panel
p1 + theme_oat()

# Cool grey panel
p1 + theme_oat(panel_background_fill = "#f2f2f2ff")
```

---

theme_stone *Stone theme*

Description

A complete theme with a stone grey panel 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_stone(
  ...,
  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 = "#EBEBEBFF",
  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)

p1 <- penguins |>
  ggplot(aes(x = species, y = body_mass, colour = species, fill = species)) +
  geom_jitter(shape = 21) +
  scale_colour_discrete(palette = blends::multiply(scales::pal_hue()))
```

```
# Default: flexoki base50 oat panel
p1 + theme_oat()

# Cool grey panel
p1 + theme_oat(panel_background_fill = "#f2f2f2ff")
```

`theme_white` *White theme*

Description

A complete theme for a white panel background.

Usage

```
theme_white(
  ...,
  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)
library(ggrefine)

p_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_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_white  <- p_light + theme_white() + labs(title = "theme_white")
p_oat    <- p_light + theme_oat()   + labs(title = "theme_oat")
p_stone <- p_light + theme_stone() + labs(title = 'theme_stone')
```

```
p_black  <- p_dark  + theme_black() + labs(title = "theme_black")

patchwork::wrap_plots(
  p_white,
  p_black,
  p_oat,
  p_stone
)
```

Index