

Package ‘ggrefine’

May 8, 2026

Title Publication-Quality 'ggplot2' Themes

Version 0.4.0

Description Complete themes for publication-quality 'ggplot2' visualisation.
Also provides functions to modify these based on the positional axis scales
and focus of a particular 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 knitr, patchwork, rmarkdown, spelling

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 2026-05-04 13:10:10 UTC

Contents

classic	2
hybrid	3
minimal	4
modern	6
none	7

theme_dark	9
theme_grey	11
theme_light	14
theme_oat	17
void	20

Index 22

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

Description

Removes gridlines and ticks from discrete axes.

Usage

```
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 = ggrefine::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::classic() + labs(title = "ggrefine::classic"),
  p_discrete_x + ggrefine::classic(x_type = "discrete"),
  p_discrete_y + ggrefine::classic(y_type = "discrete"),
  p_continuous + ggrefine::modern() + labs(title = "ggrefine::modern"),
  p_discrete_x + ggrefine::modern(x_type = "discrete"),
  p_discrete_y + ggrefine::modern(y_type = "discrete"),
  p_continuous + ggrefine::hybrid() + labs(title = "ggrefine::hybrid"),
  p_discrete_x + ggrefine::hybrid(x_type = "discrete"),
  p_discrete_y + ggrefine::hybrid(y_type = "discrete"),
  p_continuous + ggrefine::minimal() + labs(title = "ggrefine::minimal"),
  p_discrete_x + ggrefine::minimal(x_type = "discrete"),
  p_discrete_y + ggrefine::minimal(y_type = "discrete"),
  p_continuous + ggrefine::void() + labs(title = "ggrefine::void"),
  p_discrete_x + ggrefine::void(x_type = "discrete"),
  p_discrete_y + ggrefine::void(y_type = "discrete"),
  p_continuous + ggrefine::none() + labs(title = "ggrefine::none"),
  p_discrete_x + ggrefine::none(x_type = "discrete"),
  p_discrete_y + ggrefine::none(y_type = "discrete"),
  ncol = 3
)

```

 hybrid

Hybrid refine

Description

Behaves like `classic()` when both axes are continuous or binned — leaving gridlines and axis elements untouched. When a discrete axis is present, behaves like `modern()`, removing gridlines and axis line/tick elements from the non-focused dimension and stripping ticks from the discrete axis.

Usage

```
hybrid(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 <code>NULL</code> (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 = ggrefine::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::classic() + labs(title = "ggrefine::classic"),
  p_discrete_x + ggrefine::classic(x_type = "discrete"),
  p_discrete_y + ggrefine::classic(y_type = "discrete"),
  p_continuous + ggrefine::modern() + labs(title = "ggrefine::modern"),
  p_discrete_x + ggrefine::modern(x_type = "discrete"),
  p_discrete_y + ggrefine::modern(y_type = "discrete"),
  p_continuous + ggrefine::hybrid() + labs(title = "ggrefine::hybrid"),
  p_discrete_x + ggrefine::hybrid(x_type = "discrete"),
  p_discrete_y + ggrefine::hybrid(y_type = "discrete"),
  p_continuous + ggrefine::minimal() + labs(title = "ggrefine::minimal"),
  p_discrete_x + ggrefine::minimal(x_type = "discrete"),
  p_discrete_y + ggrefine::minimal(y_type = "discrete"),
  p_continuous + ggrefine::void() + labs(title = "ggrefine::void"),
  p_discrete_x + ggrefine::void(x_type = "discrete"),
  p_discrete_y + ggrefine::void(y_type = "discrete"),
  p_continuous + ggrefine::none() + labs(title = "ggrefine::none"),
  p_discrete_x + ggrefine::none(x_type = "discrete"),
  p_discrete_y + ggrefine::none(y_type = "discrete"),
  ncol = 3
)
```

Description

Behaves like `hybrid()` but additionally removes axis lines and ticks from both axes. When both axes are continuous or binned, only axis lines and ticks are removed, leaving gridlines untouched. When a discrete axis is present, also removes gridlines and axis elements from the non-focused dimension, as in `modern()`.

Usage

```
minimal(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 = ggrefine::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::classic() + labs(title = "ggrefine::classic"),
  p_discrete_x + ggrefine::classic(x_type = "discrete"),
  p_discrete_y + ggrefine::classic(y_type = "discrete"),
  p_continuous + ggrefine::modern() + labs(title = "ggrefine::modern"),
  p_discrete_x + ggrefine::modern(x_type = "discrete"),
  p_discrete_y + ggrefine::modern(y_type = "discrete"),
  p_continuous + ggrefine::hybrid() + labs(title = "ggrefine::hybrid"),
```

```

p_discrete_x + ggrefine::hybrid(x_type = "discrete"),
p_discrete_y + ggrefine::hybrid(y_type = "discrete"),
p_continuous + ggrefine::minimal() + labs(title = "ggrefine::minimal"),
p_discrete_x + ggrefine::minimal(x_type = "discrete"),
p_discrete_y + ggrefine::minimal(y_type = "discrete"),
p_continuous + ggrefine::void() + labs(title = "ggrefine::void"),
p_discrete_x + ggrefine::void(x_type = "discrete"),
p_discrete_y + ggrefine::void(y_type = "discrete"),
p_continuous + ggrefine::none() + labs(title = "ggrefine::none"),
p_discrete_x + ggrefine::none(x_type = "discrete"),
p_discrete_y + ggrefine::none(y_type = "discrete"),
ncol = 3
)

```

modern

Modern refine

Description

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

Usage

```
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 = ggrefine::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::classic() + labs(title = "ggrefine::classic"),
  p_discrete_x + ggrefine::classic(x_type = "discrete"),
  p_discrete_y + ggrefine::classic(y_type = "discrete"),
  p_continuous + ggrefine::modern() + labs(title = "ggrefine::modern"),
  p_discrete_x + ggrefine::modern(x_type = "discrete"),
  p_discrete_y + ggrefine::modern(y_type = "discrete"),
  p_continuous + ggrefine::hybrid() + labs(title = "ggrefine::hybrid"),
  p_discrete_x + ggrefine::hybrid(x_type = "discrete"),
  p_discrete_y + ggrefine::hybrid(y_type = "discrete"),
  p_continuous + ggrefine::minimal() + labs(title = "ggrefine::minimal"),
  p_discrete_x + ggrefine::minimal(x_type = "discrete"),
  p_discrete_y + ggrefine::minimal(y_type = "discrete"),
  p_continuous + ggrefine::void() + labs(title = "ggrefine::void"),
  p_discrete_x + ggrefine::void(x_type = "discrete"),
  p_discrete_y + ggrefine::void(y_type = "discrete"),
  p_continuous + ggrefine::none() + labs(title = "ggrefine::none"),
  p_discrete_x + ggrefine::none(x_type = "discrete"),
  p_discrete_y + ggrefine::none(y_type = "discrete"),
  ncol = 3
)

```

none

No refine

Description

Leaves the theme unchanged.

Usage

```
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/binning y gives "x", continuous/binning x with discrete y gives "y", otherwise "x".

... Additional arguments (currently unused).

Value

An empty `ggplot2` theme object

Examples

```
library(ggplot2)

set_theme(new = ggrefine::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::classic() + labs(title = "ggrefine::classic"),
  p_discrete_x + ggrefine::classic(x_type = "discrete"),
  p_discrete_y + ggrefine::classic(y_type = "discrete"),
  p_continuous + ggrefine::modern() + labs(title = "ggrefine::modern"),
  p_discrete_x + ggrefine::modern(x_type = "discrete"),
  p_discrete_y + ggrefine::modern(y_type = "discrete"),
  p_continuous + ggrefine::hybrid() + labs(title = "ggrefine::hybrid"),
  p_discrete_x + ggrefine::hybrid(x_type = "discrete"),
  p_discrete_y + ggrefine::hybrid(y_type = "discrete"),
  p_continuous + ggrefine::minimal() + labs(title = "ggrefine::minimal"),
  p_discrete_x + ggrefine::minimal(x_type = "discrete"),
  p_discrete_y + ggrefine::minimal(y_type = "discrete"),
  p_continuous + ggrefine::void() + labs(title = "ggrefine::void"),
  p_discrete_x + ggrefine::void(x_type = "discrete"),
  p_discrete_y + ggrefine::void(y_type = "discrete"),
  p_continuous + ggrefine::none() + labs(title = "ggrefine::none"),
  p_discrete_x + ggrefine::none(x_type = "discrete"),
  p_discrete_y + ggrefine::none(y_type = "discrete"),
  ncol = 3
)
```

theme_dark	<i>Dark theme</i>
------------	-------------------

Description

A complete theme for a dark plot and panel background. The plot background and panel grid default to "black".

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",
  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 length of 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.
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")
p_oat <- p_base_light + ggrefine::theme_oat() + labs(title = "ggrefine::theme_oat")

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

theme_grey

Grey theme

Description

A complete theme for a grey panel background on a white plot background. The panel background fill defaults to "grey92". The default panel grid colour is derived automatically by blending the panel_background_fill with itself using blends::multiply() to produce a darker tone that stays harmonious with the panel background.

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 = "grey92",
  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",
  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 length of 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.
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")
p_oat <- p_base_light + ggrefine::theme_oat() + labs(title = "ggrefine::theme_oat")

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

```

 theme_light

Light theme

Description

A complete theme for a white plot and 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",
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 length of 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.
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")
p_oat <- p_base_light + ggrefine::theme_oat() + labs(title = "ggrefine::theme_oat")

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

```

theme_oat

Oat theme

Description

A complete theme for a oat panel background on a white plot background. The panel background fill defaults so `flexoki::flexoki$base["base50"]`. The default panel grid colour is derived automatically by blending the `panel_background_fill` with itself using `blends::multiply()` to produce a darker tone that stays harmonious with the panel background.

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",

```

```

    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 length of 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.
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")
p_oat <- p_base_light + ggrefine::theme_oat() + labs(title = "ggrefine::theme_oat")

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

 void

Void refine

Description

Removes axes and gridlines.

Usage

```
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 = ggrefine::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::classic() + labs(title = "ggrefine::classic"),
  p_discrete_x + ggrefine::classic(x_type = "discrete"),
  p_discrete_y + ggrefine::classic(y_type = "discrete"),
  p_continuous + ggrefine::modern() + labs(title = "ggrefine::modern"),
```

```
p_discrete_x + ggrefine::modern(x_type = "discrete"),
p_discrete_y + ggrefine::modern(y_type = "discrete"),
p_continuous + ggrefine::hybrid() + labs(title = "ggrefine::hybrid"),
p_discrete_x + ggrefine::hybrid(x_type = "discrete"),
p_discrete_y + ggrefine::hybrid(y_type = "discrete"),
p_continuous + ggrefine::minimal() + labs(title = "ggrefine::minimal"),
p_discrete_x + ggrefine::minimal(x_type = "discrete"),
p_discrete_y + ggrefine::minimal(y_type = "discrete"),
p_continuous + ggrefine::void() + labs(title = "ggrefine::void"),
p_discrete_x + ggrefine::void(x_type = "discrete"),
p_discrete_y + ggrefine::void(y_type = "discrete"),
p_continuous + ggrefine::none() + labs(title = "ggrefine::none"),
p_discrete_x + ggrefine::none(x_type = "discrete"),
p_discrete_y + ggrefine::none(y_type = "discrete"),
ncol = 3
)
```

Index

classic, [2](#)
classic(), [3](#)

hybrid, [3](#)
hybrid(), [5](#)

minimal, [4](#)
modern, [6](#)
modern(), [3](#), [5](#)

none, [7](#)

theme_dark, [9](#)
theme_grey, [11](#)
theme_light, [14](#)
theme_oat, [17](#)

void, [20](#)