

Package ‘ggscribe’

May 2, 2026

Title Publication-Quality 'ggplot2' Annotation

Version 0.1.0

Description Annotation helper functions for publication-quality 'ggplot2' visualisation. These functions make it easier to annotate plots in a way that stays consistent with the set theme.

License MIT + file LICENSE

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

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

Depends R (>= 4.1.0)

Imports farver, ggplot2 (>= 4.0.0), glue, grid, rlang, scales

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Suggests blends, dplyr, flexoki, ggrefine, ggwidth, jumble, knitr,
rmarkdown, spelling, stringr

Config/testthat/edition 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 2026-05-02 09:20:02 UTC

Contents

aes_contrast	2
axis_bracket	4
axis_line	6

axis_text	7
axis_ticks	9
guide_sec_axis_text	10
panel_shade	12
reference_line	14
sec_axis_text	15
theme_sec_axis_text	17

Index 19

aes_contrast	<i>A mapped aesthetic for text colour on fill</i>
--------------	---

Description

Modifies a mapped colour (or fill) aesthetic for contrast against the fill (or colour) aesthetic.

Function can be spliced into `ggplot2::aes` with `rlang::!!!`.

Usage

```
aes_contrast(..., dark = NULL, light = NULL, aesthetic = "colour")
```

Arguments

...	Require named arguments (and support trailing commas).
dark	A dark colour. If NULL, derived from theme text or panel background.
light	A light colour. If NULL, derived from theme text or panel background.
aesthetic	The aesthetic to be modified for contrast. Either "colour" (default) or "fill".

Value

A ggplot2 aesthetic in `ggplot2::aes`.

See Also

[splice](#)

Examples

```
library(ggplot2)
library(dplyr)
library(stringr)

set_theme(
  ggrepine::theme_light(
    panel_heights = rep(unit(50, "mm"), 100),
    panel_widths = rep(unit(75, "mm"), 100),
  )
)
```

```

)

ggwidth::set_equiwidth(equiwidth = 1.75)

mtcars |>
  count(cyl, am) |>
  mutate(
    am = if_else(am == 0, "Automatic", "Manual"),
    cyl = as.factor(cyl)
  ) |>
  ggplot(aes(x = am, y = n, colour = cyl, fill = cyl, label = n)) +
  geom_col(
    position = position_dodge2(preserve = "single", padding = 0.05),
    width = ggwidth::get_width(n = 2, n_dodge = 3),
  ) +
  scale_fill_discrete(palette = jumble::jumble) +
  scale_colour_discrete(palette = blends::multiply(jumble::jumble)) +
  geom_text(
    mapping = ggscribe::aes_contrast(), # or aes(!!!ggscribe::aes_contrast()),
    position = position_dodge2(
      width = ggwidth::get_width(n = 2, n_dodge = 3),
      padding = 0.05,
      preserve = "single"),
    vjust = 1.33,
    show.legend = FALSE,
  ) +
  scale_y_continuous(expand = expansion(c(0, 0.05))) +
  ggrefine::modern(x_type = "discrete")

mtcars |>
  count(cyl, am) |>
  mutate(
    am = if_else(am == 0, "automatic", "manual"),
    am = stringr::str_to_sentence(am),
    cyl = as.factor(cyl)
  ) |>
  ggplot(aes(y = am, x = n, colour = cyl, fill = cyl, label = n)) +
  geom_col(
    position = position_dodge2(preserve = "single", padding = 0.05),
    width = ggwidth::get_width(n = 2, n_dodge = 3, orientation = "y"),
  ) +
  scale_fill_discrete(palette = jumble::jumble) +
  scale_colour_discrete(palette = blends::multiply(jumble::jumble)) +
  geom_text(
    mapping = ggscribe::aes_contrast(), # or aes(!!!ggscribe::aes_contrast()),
    position = position_dodge2(
      width = ggwidth::get_width(n = 2, n_dodge = 3, orientation = "y"),
      preserve = "single",
      padding = 0.05,
    ),
    hjust = 1.25,
    show.legend = FALSE,
  ) +

```

```
scale_x_continuous(expand = expansion(c(0, 0.05))) +
ggrefine::modern(y_type = "discrete")
```

axis_bracket

Annotate an axis bracket

Description

Draws a bracket spanning `min(breaks)` to `max(breaks)` along an axis edge or at a floating data position. The bar uses the same rendering path as `axis_line()`; the caps use the same path as `axis_ticks()`. Requires `coord_cartesian(clip = "off")`.

Usage

```
axis_bracket(
  ...,
  position = NULL,
  xintercept = NULL,
  yintercept = NULL,
  breaks,
  colour = NULL,
  linewidth = NULL,
  linetype = NULL,
  length = ggplot2::rel(1)
)
```

Arguments

<code>...</code>	Not used. Forces named arguments.
<code>position</code>	One of "top", "bottom", "left", or "right". Inferred from <code>xintercept</code> or <code>yintercept</code> if not provided.
<code>xintercept</code>	For "left"/"right" axes: float the bracket to this x position in data coordinates instead of the panel edge.
<code>yintercept</code>	For "top"/"bottom" axes: float the bracket to this y position in data coordinates instead of the panel edge.
<code>breaks</code>	A numeric vector of length ≥ 2 . The bar spans <code>min(breaks)</code> to <code>max(breaks)</code> ; caps are drawn at every break value.
<code>colour</code>	Inherits from <code>axis.ticks</code> in the set theme (falling back through <code>axis.line</code> and <code>line</code>).
<code>linewidth</code>	Inherits from <code>axis.ticks</code> in the set theme. Supports <code>rel()</code> .
<code>linetype</code>	Inherits from <code>axis.ticks</code> in the set theme.
<code>length</code>	Length of the bracket caps as a grid unit. Supports <code>rel()</code> . Negative values flip the cap direction. Defaults to <code>rel(1)</code> (outward at theme tick length).

Value

A list of ggplot2 annotation layers.

See Also

[axis_line\(\)](#), [axis_ticks\(\)](#), [axis_text\(\)](#), [reference_line\(\)](#), [panel_shade\(\)](#), [sec_axis_text\(\)](#)

Examples

```
library(ggplot2)
library(dplyr)

set_theme(
  ggrefine::theme_grey(
    panel_heights = rep(unit(50, "mm"), 100),
    panel_widths = rep(unit(75, "mm"), 100),
  )
)

mtcars |>
  ggplot(aes(x = wt, y = mpg, colour = as.factor(gear), fill = as.factor(gear))) +
  scale_colour_discrete(palette = blends::multiply(get_theme()$palette.colour.discrete)) +
  #clip = "off" is required for axis_text, axis_ticks and axis_bracket
  coord_cartesian(clip = "off") +
  #reference lines and shade
  ggscribe::reference_line(xintercept = 2.4) +
  ggscribe::reference_line(yintercept = 12) +
  ggscribe::panel_shade(
    xmin = 4,
    xmax = 5,
  ) +
  #top axis
  scale_x_continuous(
    sec.axis = ggscribe::sec_axis_text(
      breaks = c(mean(c(4, 5))),
      labels = c("Range"),
      guide = ggscribe::guide_sec_axis_text(
        angle = 90,
      )
    )
  ) +
  ggscribe::axis_bracket(
    position = "top",
    breaks = c(4, 5),
  ) +
  ggscribe::axis_text(
    position = "top",
    breaks = c(2.4),
    labels = c("Threshold"),
  ) +
  #right axis
  ggscribe::axis_text(
```

```

    position = "right",
    breaks = 12,
    labels = "Threshold",
  ) +
  #'geom
  geom_point() +
  #annotations fit plot
  theme(plot.background = element_rect(colour = "grey92"))

```

axis_line

Annotate an axis line

Description

Draws a line along an axis edge, with style defaults taken from the `axis.line` element of the set theme. Requires `coord_cartesian(clip = "off")`.

Usage

```

axis_line(
  ...,
  position = NULL,
  xintercept = NULL,
  yintercept = NULL,
  colour = NULL,
  linewidth = NULL,
  linetype = NULL
)

```

Arguments

...	Not used. Forces named arguments.
position	One of "top", "bottom", "left", or "right". Inferred from <code>xintercept</code> or <code>yintercept</code> if not provided.
xintercept	For "left"/"right" axes: float the axis to this x position in data coordinates instead of the panel edge.
yintercept	For "top"/"bottom" axes: float the axis to this y position in data coordinates instead of the panel edge.
colour	Inherits from <code>axis.line</code> in the set theme.
linewidth	Inherits from <code>axis.line</code> in the set theme. Supports <code>rel()</code> .
linetype	Inherits from <code>axis.line</code> in the set theme.

Value

A list of `ggplot2` annotation layers.

See Also

[axis_ticks\(\)](#), [axis_text\(\)](#), [axis_bracket\(\)](#), [reference_line\(\)](#), [panel_shade\(\)](#), [sec_axis_text\(\)](#)

axis_text	<i>Annotate axis text</i>
-----------	---------------------------

Description

Draws text labels at specified break positions along an axis, with style defaults taken from the `axis.text` element of the set theme. Requires `coord_cartesian(clip = "off")`.

Usage

```
axis_text(
  ...,
  position = NULL,
  xintercept = NULL,
  yintercept = NULL,
  breaks,
  labels = NULL,
  colour = NULL,
  size = NULL,
  family = NULL,
  hjust = NULL,
  vjust = NULL,
  angle = 0,
  length = ggplot2::rel(1)
)
```

Arguments

...	Not used. Forces named arguments.
position	One of "top", "bottom", "left", or "right". Inferred from <code>xintercept</code> or <code>yintercept</code> if not provided.
xintercept	For "left"/"right" axes: float the axis to this x position in data coordinates instead of the panel edge.
yintercept	For "top"/"bottom" axes: float the axis to this y position in data coordinates instead of the panel edge.
breaks	A numeric vector of break positions.
labels	One of: <ul style="list-style-type: none"> • NULL (default) to use break values as labels • A character vector the same length as <code>breaks</code> • A function taking break values and returning labels
colour	Inherits from <code>axis.text</code> in the set theme.

size	Inherits from <code>axis.text</code> in the set theme.
family	Inherits from <code>axis.text</code> in the set theme.
hjust, vjust	Justification. Auto-calculated from position if NULL.
angle	Text rotation angle. Defaults to 0.
length	Offset from the axis edge including tick length and margin. Supports <code>rel()</code> . Negative values place labels inside the panel. Defaults to <code>rel(1)</code> (theme tick length + text margin).

Value

A list of `ggplot2` annotation layers.

See Also

[axis_line\(\)](#), [axis_ticks\(\)](#), [axis_bracket\(\)](#), [reference_line\(\)](#), [panel_shade\(\)](#), [sec_axis_text\(\)](#)

Examples

```
library(ggplot2)
library(dplyr)

set_theme(
  ggrefine::theme_grey(
    panel_heights = rep(unit(50, "mm"), 100),
    panel_widths = rep(unit(75, "mm"), 100),
  )
)

mtcars |>
  ggplot(aes(x = wt, y = mpg, colour = as.factor(gear), fill = as.factor(gear))) +
  scale_colour_discrete(palette = blends::multiply(get_theme()$palette.colour.discrete)) +
  #clip = "off" is required for axis_text, axis_ticks and axis_bracket
  coord_cartesian(clip = "off") +
  #reference lines and shade
  ggscribe::reference_line(xintercept = 2.4) +
  ggscribe::reference_line(yintercept = 12) +
  ggscribe::panel_shade(
    xmin = 4,
    xmax = 5,
  ) +
  #top axis
  scale_x_continuous(
    sec.axis = ggscribe::sec_axis_text(
      breaks = c(mean(c(4, 5))),
      labels = c("Range"),
      guide = ggscribe::guide_sec_axis_text(
        angle = 90,
      )
    )
  ) +
  ggscribe::axis_bracket(
```

```

    position = "top",
    breaks = c(4, 5),
  ) +
  ggscribe::axis_text(
    position = "top",
    breaks = c(2.4),
    labels = c("Threshold"),
  ) +
  #right axis
  ggscribe::axis_text(
    position = "right",
    breaks = 12,
    labels = "Threshold",
  ) +
  #'geom
  geom_point() +
  #annotations fit plot
  theme(plot.background = element_rect(colour = "grey92"))

```

axis_ticks

*Annotate axis ticks***Description**

Draws axis ticks at specified break positions, with style defaults taken from the `axis_ticks` element of the set theme. Requires `coord_cartesian(clip = "off")`.

Usage

```

axis_ticks(
  ...,
  position = NULL,
  xintercept = NULL,
  yintercept = NULL,
  breaks,
  minor = FALSE,
  colour = NULL,
  linewidth = NULL,
  length = ggplot2::rel(1)
)

```

Arguments

...	Not used. Forces named arguments.
position	One of "top", "bottom", "left", or "right". Inferred from xintercept or yintercept if not provided.
xintercept	For "left"/"right" axes: float the axis to this x position in data coordinates instead of the panel edge.

yintercept	For "top"/"bottom" axes: float the axis to this y position in data coordinates instead of the panel edge.
breaks	A numeric vector of break positions.
minor	Logical. If TRUE, uses minor tick theme defaults. Defaults to FALSE.
colour	Inherits from <code>axis.ticks</code> in the set theme.
linewidth	Inherits from <code>axis.ticks</code> in the set theme. Supports <code>rel()</code> .
length	Total tick length as a grid unit. Supports <code>rel()</code> . Negative values flip the tick direction (inward). Defaults to <code>rel(1)</code> (outward at theme tick length).

Value

A list of `ggplot2` annotation layers.

See Also

[axis_line\(\)](#), [axis_text\(\)](#), [axis_bracket\(\)](#), [reference_line\(\)](#), [panel_shade\(\)](#), [sec_axis_text\(\)](#)

`guide_sec_axis_text` *Guide optimised for secondary axis text annotations*

Description

A wrapper around `ggplot2::guide_axis()` that defaults to using `theme_sec_axis_text()`. This guide is designed to strip away standard axis furniture (like lines and ticks) while preserving text, making it ideal for secondary axes used as margin labels.

Usage

```
guide_sec_axis_text(..., theme = theme_sec_axis_text())
```

Arguments

<code>...</code>	Additional arguments passed to <code>ggplot2::guide_axis()</code> , such as <code>title</code> , <code>check.overlap</code> , or <code>angle</code> .
<code>theme</code>	A theme object to style the guide. Defaults to <code>theme_sec_axis_text()</code> , which suppresses ticks and lines.

Value

A guide object to be used in a scale's guide argument or within `sec_axis_text()`.

See Also

[sec_axis_text\(\)](#), [theme_sec_axis_text\(\)](#)

Examples

```

library(ggplot2)
library(dplyr)

set_theme(
  ggrefine::theme_grey(
    panel_heights = rep(unit(50, "mm"), 100),
    panel_widths = rep(unit(75, "mm"), 100),
  )
)

mtcars |>
  ggplot(aes(x = wt, y = mpg, colour = as.factor(gear), fill = as.factor(gear))) +
  scale_colour_discrete(palette = blends::multiply(get_theme())$palette.colour.discrete) +
  #clip = "off" is required for axis_text, axis_ticks and axis_bracket
  coord_cartesian(clip = "off") +
  #reference lines and shade
  ggscribe::reference_line(xintercept = 2.4) +
  ggscribe::reference_line(yintercept = 12) +
  ggscribe::panel_shade(
    xmin = 4,
    xmax = 5,
  ) +
  #top axis
  scale_x_continuous(
    sec.axis = ggscribe::sec_axis_text(
      breaks = c(mean(c(4, 5))),
      labels = c("Range"),
      guide = ggscribe::guide_sec_axis_text(
        angle = 90,
      )
    )
  ) +
  ggscribe::axis_bracket(
    position = "top",
    breaks = c(4, 5),
  ) +
  ggscribe::axis_text(
    position = "top",
    breaks = c(2.4),
    labels = c("Threshold"),
  ) +
  #right axis
  ggscribe::axis_text(
    position = "right",
    breaks = 12,
    labels = "Threshold",
  ) +
  #'geom
  geom_point() +
  #annotations fit plot
  theme(plot.background = element_rect(colour = "grey92"))

```

panel_shade	<i>Annotate a shaded panel region</i>
-------------	---------------------------------------

Description

Draws a filled rectangle over the panel with colour defaults taken from the set theme. Defaults to a subtle overlay across the full panel, with the fill automatically adapting to light or dark panel backgrounds. Should be placed before geom layers.

Usage

```
panel_shade(
  ...,
  xmin = -Inf,
  xmax = Inf,
  ymin = -Inf,
  ymax = Inf,
  fill = "#878580",
  alpha = 0.25,
  colour = "transparent",
  linewidth = NULL,
  linetype = NULL
)
```

Arguments

...	Not used. Allows trailing commas and named-argument style calls.
xmin, xmax	Left and right edges of the rectangle. Defaults to -Inf and Inf. Use I() for normalized coordinates (0-1).
ymin, ymax	Bottom and top edges of the rectangle. Defaults to -Inf and Inf. Use I() for normalized coordinates (0-1).
fill	Fill colour. Defaults to a neutral grey.
alpha	Opacity of the rectangle. Defaults to 0.25.
colour	Border colour. Defaults to "transparent".
linewidth	Inherits from panel.border in the set theme. Supports rel().
linetype	Border linetype. Defaults to 1.

Value

A list containing an annotation layer.

Examples

```

library(ggplot2)
library(dplyr)

set_theme(
  ggrefine::theme_grey(
    panel_heights = rep(unit(50, "mm"), 100),
    panel_widths = rep(unit(75, "mm"), 100),
  )
)

mtcars |>
  ggplot(aes(x = wt, y = mpg, colour = as.factor(gear), fill = as.factor(gear))) +
  scale_colour_discrete(palette = blends::multiply(get_theme())$palette.colour.discrete) +
  #clip = "off" is required for axis_text, axis_ticks and axis_bracket
  coord_cartesian(clip = "off") +
  #reference lines and shade
  ggscribe::reference_line(xintercept = 2.4) +
  ggscribe::reference_line(yintercept = 12) +
  ggscribe::panel_shade(
    xmin = 4,
    xmax = 5,
  ) +
  #top axis
  scale_x_continuous(
    sec.axis = ggscribe::sec_axis_text(
      breaks = c(mean(c(4, 5))),
      labels = c("Range"),
      guide = ggscribe::guide_sec_axis_text(
        angle = 90,
      )
    )
  ) +
  ggscribe::axis_bracket(
    position = "top",
    breaks = c(4, 5),
  ) +
  ggscribe::axis_text(
    position = "top",
    breaks = c(2.4),
    labels = c("Threshold"),
  ) +
  #right axis
  ggscribe::axis_text(
    position = "right",
    breaks = 12,
    labels = "Threshold",
  ) +
  #'geom
  geom_point() +
  #annotations fit plot
  theme(plot.background = element_rect(colour = "grey92"))

```

reference_line	<i>Annotate a reference line</i>
----------------	----------------------------------

Description

Draws a reference line within the panel, with style defaults taken from the `axis.line` element of the set theme.

Usage

```
reference_line(  
  ...,  
  xintercept = NULL,  
  yintercept = NULL,  
  colour = NULL,  
  linewidth = NULL,  
  linetype = "dashed"  
)
```

Arguments

<code>...</code>	Not used. Forces named arguments.
<code>xintercept</code>	Draw a vertical reference line at this x position.
<code>yintercept</code>	Draw a horizontal reference line at this y position.
<code>colour</code>	Inherits from <code>axis.line</code> in the set theme.
<code>linewidth</code>	Inherits from <code>axis.line</code> in the set theme. Supports <code>rel()</code> .
<code>linetype</code>	Defaults to "dashed".

Value

A list of ggplot2 annotation layers.

See Also

[axis_line\(\)](#), [axis_ticks\(\)](#), [axis_text\(\)](#), [axis_bracket\(\)](#), [panel_shade\(\)](#), [sec_axis_text\(\)](#)

Examples

```
library(ggplot2)  
library(dplyr)  
  
set_theme(  
  ggrefine::theme_grey(  
    panel_heights = rep(unit(50, "mm"), 100),  
    panel_widths = rep(unit(75, "mm"), 100),
```

```

)
)

mtcars |>
  ggplot(aes(x = wt, y = mpg, colour = as.factor(gear), fill = as.factor(gear))) +
  scale_colour_discrete(palette = blends::multiply(get_theme())$palette.colour.discrete)) +
  #clip = "off" is required for axis_text, axis_ticks and axis_bracket
  coord_cartesian(clip = "off") +
  #reference lines and shade
  ggscribe::reference_line(xintercept = 2.4) +
  ggscribe::reference_line(yintercept = 12) +
  ggscribe::panel_shade(
    xmin = 4,
    xmax = 5,
  ) +
  #top axis
  scale_x_continuous(
    sec.axis = ggscribe::sec_axis_text(
      breaks = c(mean(c(4, 5))),
      labels = c("Range"),
      guide = ggscribe::guide_sec_axis_text(
        angle = 90,
      )
    )
  ) +
  ggscribe::axis_bracket(
    position = "top",
    breaks = c(4, 5),
  ) +
  ggscribe::axis_text(
    position = "top",
    breaks = c(2.4),
    labels = c("Threshold"),
  ) +
  #right axis
  ggscribe::axis_text(
    position = "right",
    breaks = 12,
    labels = "Threshold",
  ) +
  #'geom
  geom_point() +
  #annotations fit plot
  theme(plot.background = element_rect(colour = "grey92"))

```

sec_axis_text

Secondary axis optimised for text annotations

Description

Secondary axis optimised for text annotations

Usage

```
sec_axis_text(
  breaks = ggplot2::waiver(),
  labels = ggplot2::derive(),
  name = NULL,
  guide = ggplot2::guide_axis(theme = theme_sec_axis_text()),
  ...
)
```

Arguments

breaks	One of: <ul style="list-style-type: none"> • NULL for no breaks • <code>ggplot2::waiver()</code> (default) to inherit breaks from the primary axis • A numeric vector of break positions • A function that takes the scale limits as input and returns break positions (e.g. <code>\(x) mean(c(x[2], 32))</code>)
labels	One of: <ul style="list-style-type: none"> • <code>ggplot2::derive()</code> (default) to derive labels from breaks • A character vector of labels, the same length as breaks • A function that takes break positions as input and returns labels
name	The name of the secondary axis. Use <code>ggplot2::waiver()</code> to derive the name from the primary axis, or NULL (default) for no name.
guide	A guide object used to render the axis. Defaults to <code>guide_sec_axis_text()</code> , which uses <code>theme_sec_axis_text()</code> to make transparent ticks and lines by default.
...	Additional arguments passed to <code>ggplot2::dup_axis()</code> .

Value

A `AxisSecondary` object for use in the `sec.axis` argument of `scale_x_continuous()` or `scale_y_continuous()`.

See Also

[guide_sec_axis_text\(\)](#), [theme_sec_axis_text\(\)](#), [axis_text\(\)](#)

Examples

```
library(ggplot2)
library(dplyr)

set_theme(
  ggrefine::theme_grey(
    panel_heights = rep(unit(50, "mm"), 100),
    panel_widths = rep(unit(75, "mm"), 100),
  )
)
```

```

mtcars |>
  ggplot(aes(x = wt, y = mpg, colour = as.factor(gear), fill = as.factor(gear))) +
  scale_colour_discrete(palette = blends::multiply(get_theme()$palette.colour.discrete)) +
  #clip = "off" is required for axis_text, axis_ticks and axis_bracket
  coord_cartesian(clip = "off") +
  #reference lines and shade
  ggscribe::reference_line(xintercept = 2.4) +
  ggscribe::reference_line(yintercept = 12) +
  ggscribe::panel_shade(
    xmin = 4,
    xmax = 5,
  ) +
  #top axis
  scale_x_continuous(
    sec.axis = ggscribe::sec_axis_text(
      breaks = c(mean(c(4, 5))),
      labels = c("Range"),
      guide = ggscribe::guide_sec_axis_text(
        angle = 90,
      )
    )
  ) +
  ggscribe::axis_bracket(
    position = "top",
    breaks = c(4, 5),
  ) +
  ggscribe::axis_text(
    position = "top",
    breaks = c(2.4),
    labels = c("Threshold"),
  ) +
  #right axis
  ggscribe::axis_text(
    position = "right",
    breaks = 12,
    labels = "Threshold",
  ) +
  #'geom
  geom_point() +
  #annotations fit plot
  theme(plot.background = element_rect(colour = "grey92"))

```

theme_sec_axis_text *Theme adjustments optimised for secondary axis text annotations*

Description

Theme adjustments optimised for secondary axis text annotations

Usage

```
theme_sec_axis_text(  
  axis = NULL,  
  axis_ticks_to = "transparent",  
  axis_line_to = "transparent",  
  axis_text_to = "keep",  
  axis_title_to = "keep"  
)
```

Arguments

<code>axis</code>	Character. "x", "y", or NULL (defaults to both).
<code>axis_ticks_to</code>	Action for ticks: "transparent", "blank", or "keep".
<code>axis_line_to</code>	Action for lines: "transparent", "blank", or "keep".
<code>axis_text_to</code>	Action for text: "transparent", "blank", or "keep".
<code>axis_title_to</code>	Action for titles: "transparent", "blank", or "keep".

Value

A ggplot2 theme object.

See Also

[sec_axis_text\(\)](#), [guide_sec_axis_text\(\)](#)
[axis_ticks\(\)](#), [axis_line\(\)](#), [axis_text\(\)](#), [reference_line\(\)](#)

Index

aes_contrast, [2](#)
axis_bracket, [4](#)
axis_bracket(), [7](#), [8](#), [10](#), [14](#)
axis_line, [6](#)
axis_line(), [4](#), [5](#), [8](#), [10](#), [14](#), [18](#)
axis_text, [7](#)
axis_text(), [5](#), [7](#), [10](#), [14](#), [16](#), [18](#)
axis_ticks, [9](#)
axis_ticks(), [4](#), [5](#), [7](#), [8](#), [14](#), [18](#)

ggplot2::aes, [2](#)
ggplot2::derive(), [16](#)
ggplot2::dup_axis(), [16](#)
ggplot2::guide_axis(), [10](#)
ggplot2::waiver(), [16](#)
guide_sec_axis_text, [10](#)
guide_sec_axis_text(), [16](#), [18](#)

panel_shade, [12](#)
panel_shade(), [5](#), [7](#), [8](#), [10](#), [14](#)

reference_line, [14](#)
reference_line(), [5](#), [7](#), [8](#), [10](#), [18](#)

sec_axis_text, [15](#)
sec_axis_text(), [5](#), [7](#), [8](#), [10](#), [14](#), [18](#)
splice, [2](#)

theme_sec_axis_text, [17](#)
theme_sec_axis_text(), [10](#), [16](#)