Package: ggscribe (via r-universe)

June 6, 2026

Title Publication-Quality 'ggplot2' Annotation

Version 0.1.1.9000

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

Roxygen list(markdown = TRUE)

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

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Repository https://davidhodge931.r-universe.dev

Date/Publication 2026-06-06 22:39:46 UTC

RemoteUrl https://github.com/davidhodge931/ggscribe

RemoteRef HEAD

RemoteSha d222989411441add96ccb82ab0d0f3359cbbb5d8

Contents

`aes_contrast` *A mapped aesthetic for text colour on fill*

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(
 ggrefine::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 one or more brackets along a floating axis line. Each bracket spans `min(breaks)` to `max(breaks)` with caps at every break value. Requires `coord_cartesian(clip = "off")`.

Usage

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

Arguments

`xintercept`	One or more x positions for vertical axis lines, in data coordinates or wrapped in `I()` for normalised panel coordinates (npc). May be a vector; each value produces a separate axis.
`yintercept`	One or more y positions for horizontal axis lines, in data coordinates or wrapped in `I()` for normalised panel coordinates (npc). May be a vector; each value produces a separate axis.
`breaks`	A numeric vector of length >= 2 in data coordinates, or wrapped in `I()` for npc. The bar spans `min(breaks)` to `max(breaks)`; caps are drawn at every break value. Pass a list the same length as the total number of axes to use different breaks per axis.

length	Length of the bracket caps. Supports rel(). Negative values flip the cap direction. Defaults to rel(1). May be a vector the same length as the number of axes.
colour	Inherits from axis.ticks in the set theme (falling back through axis.line and line). May be a vector the same length as the number of axes.
linewidth	Inherits from axis.ticks in the set theme. Supports rel(). May be a vector the same length as the number of axes.
linetype	Inherits from axis.ticks in the set theme. May be a vector the same length as the number of axes.
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Details

Caps always point in the positive direction by default (right for xintercept, up for yintercept). Use a negative length to flip them (e.g. length = -rel(1)).

Value

A list of ggplot2 annotation layers.

See Also

axis_line(), axis_ticks(), axis_text(), reference_line(), panel_shade()

---

axis_line *Annotate an axis line*

Description

Draws a full line at one or more floating positions, with style defaults taken from the axis.line element of the set theme. Requires coord_cartesian(clip = "off").

Usage

```
axis_line(
  xintercept = NULL,
  yintercept = NULL,
  colour = NULL,
  linewidth = NULL,
  linetype = NULL,
  arrow = NULL,
  layout = NULL
)
```

Arguments

xintercept	One or more x positions for vertical axis lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate line.
yintercept	One or more y positions for horizontal axis lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate line.
colour	Inherits from axis.line in the set theme. May be a vector the same length as the total number of lines.
linewidth	Inherits from axis.line in the set theme. Supports rel(). May be a vector the same length as the total number of lines.
linetype	Inherits from axis.line in the set theme. May be a vector the same length as the total number of lines.
arrow	A grid::arrow() specification, or a list the same length as the total number of lines. Must use list() not c() when supplying multiple values. E.g. grid::arrow(angle = 15, length = unit(1.5, "mm"), type = "closed").
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Details

The arrow (if any) points in the positive direction by default — rightward for xintercept lines, upward for yintercept lines.

Value

A list of ggplot2 annotation layers.

See Also

axis_ticks(), axis_text(), axis_bracket(), reference_line(), panel_shade()

---

axis_text *Annotate axis text*

Description

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

Usage

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

Arguments

xintercept	One or more x positions for vertical axis lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate axis.
yintercept	One or more y positions for horizontal axis lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate axis.
breaks	A numeric vector of break positions in data coordinates, or wrapped in I() for npc. Pass a list the same length as the total number of axes to use different breaks per axis.
labels	One of: • NULL (default) to use break values as labels • A character vector recycled across all breaks in order • A function taking break values and returning labels • A list the same length as the number of axes, each element being one of the above
length	Offset from the axis line including tick length and margin. Supports rel(). Negative values place text on the opposite side. Defaults to rel(1). May be a vector recycled across all breaks in order.
angle	Text rotation angle. Defaults to 0. May be a vector recycled across all breaks in order.
hjust, vjust	Justification. Auto-calculated from axis direction and angle if NULL. May be a vector recycled across all breaks in order.
colour	Inherits from axis.text in the set theme. May be a vector recycled across all breaks in order.
size	Inherits from axis.text in the set theme. May be a vector recycled across all breaks in order.

family	Inherits from axis.text in the set theme. May be a vector recycled across all breaks in order.
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Details

Text always sits on the positive side of the axis by default (right of xintercept lines, above yintercept lines). Use a negative length to place text on the opposite side (e.g. length = -rel(1)).

Value

A list of ggplot2 annotation layers.

See Also

axis_line(), axis_ticks(), axis_bracket(), reference_line(), panel_shade()

---

axis_ticks *Annotate axis ticks*

Description

Draws axis ticks at specified break positions along a floating axis line. Requires coord_cartesian(clip = "off").

Usage

```
axis_ticks(
  xintercept = NULL,
  yintercept = NULL,
  breaks,
  length = ggplot2::rel(1),
  colour = NULL,
  linewidth = NULL,
  linetype = NULL,
  arrow = NULL,
  layout = NULL
)
```

Arguments

xintercept	One or more x positions for vertical axis lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate axis.
yintercept	One or more y positions for horizontal axis lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate axis.
breaks	A numeric vector of break positions in data coordinates, or wrapped in I() for npc. Pass a list the same length as the total number of axes to use different breaks per axis.
length	Total tick length. Supports rel(). Negative values flip the tick direction. Defaults to rel(1). May be a vector recycled across all breaks in order.
colour	Inherits from axis.ticks in the set theme. May be a vector recycled across all breaks in order.
linewidth	Inherits from axis.ticks in the set theme. Supports rel(). May be a vector recycled across all breaks in order.
linetype	Inherits from axis.ticks in the set theme. May be a vector recycled across all breaks in order.
arrow	A grid::arrow() specification, or a list recycled across all breaks. The arrowhead points toward the axis line. Must use list() not c() when supplying multiple values. E.g. grid::arrow(angle = 15, length = unit(1.5, "mm"), type = "closed").
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Details

Ticks always point in the positive direction by default (right for xintercept, up for yintercept). Use a negative length to flip them (e.g. length = -rel(1)).

Value

A list of ggplot2 annotation layers.

See Also

axis_line(), axis_text(), axis_bracket(), reference_line(), panel_shade()

guide_sec_axis_spacer *Guide optimised for secondary axis space adjustments*

Description

A wrapper around ggplot2::guide_axis() that defaults to making transparent ticks and lines and making the text the same colour as the plot background fill from the set theme.

Usage

```
guide_sec_axis_spacer(...)
```

Arguments

...	Additional arguments passed to ggplot2::guide_axis(), such as title, check.overlap, or angle.

Value

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

See Also

sec_axis_spacer(), axis_text(), axis_ticks(), axis_bracket()

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"))
```

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

Description

A wrapper around `ggplot2::guide_axis()` that defaults to making transparent ticks and lines while preserving text, making it ideal for annotation labels.

Usage

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

Arguments

`...`	Additional arguments passed to `ggplot2::guide_axis()`, such as `title`, `check.overlap`, or `angle`.
`theme`	A theme object to style the secondary axis.

Value

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

See Also

sec_axis_text(), axis_text(), axis_ticks(), axis_bracket()

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_background` *Annotate a panel background region*

Description

Draws filled rectangles over the panel using the `panel.background` fill from the set theme as the default colour. Unlike `panel_shade()`, opacity defaults to 1 (fully opaque), making it useful for layering a solid background over existing content. Should be placed before geom layers.

Usage

```
panel_background(
  xmin = -Inf,
  xmax = Inf,
  ymin = -Inf,
  ymax = Inf,
  fill = NULL,
  alpha = 1,
  colour = NULL,
  linewidth = NULL,
  linetype = NULL,
  layout = NULL
)
```

Arguments

`xmin`, `xmax`	Left and right edges of the rectangle in data coordinates. Defaults to `-Inf` and `Inf`. Use `I()` for normalised coordinates (0-1). May be a vector for multiple rectangles.
`ymin`, `ymax`	Bottom and top edges of the rectangle in data coordinates. Defaults to `-Inf` and `Inf`. Use `I()` for normalised coordinates (0-1). May be a vector for multiple rectangles.
`fill`	Fill colour. Defaults to the `panel.background` fill from the set theme. May be a vector the same length as the bounds.
`alpha`	Opacity. Defaults to 1 (fully opaque). May be a vector.
`colour`	Border colour. Defaults to the resolved `fill` value, giving a seamless border. May be a vector.

linewidth	Inherits from panel.border in the set theme. Supports rel(). May be a vector.
linetype	Border linetype. Defaults to 1. May be a vector.
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Value

A list containing annotation layers.

See Also

panel_shade(), axis_line(), reference_line()

---

panel_grid *Annotate panel gridlines*

Description

Draws gridlines at specified positions, with style defaults taken from the panel.grid.major element of the set theme. Crop bounds (xmin, xmax, ymin, ymax) both filter which lines are drawn and control how far they run across the panel.

Usage

```
panel_grid(
  xintercept = NULL,
  yintercept = NULL,
  xmin = -Inf,
  xmax = Inf,
  ymin = -Inf,
  ymax = Inf,
  colour = NULL,
  linewidth = NULL,
  linetype = NULL,
  layout = NULL
)
```

Arguments

xintercept	One or more x positions for vertical gridlines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector.
yintercept	One or more y positions for horizontal gridlines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector.

xmin, xmax	Left and right crop bounds. Vertical gridlines outside [xmin, xmax] are not drawn; horizontal gridlines run only from xmin to xmax. Defaults to -Inf and Inf (full panel). Use I() for normalised coordinates (npc). Note: filtering (removing lines outside the range) only works when both the crop bound and the intercept are in data coordinates. npc bounds affect extent only.
ymin, ymax	Bottom and top crop bounds. Horizontal gridlines outside [ymin, ymax] are not drawn; vertical gridlines run only from ymin to ymax. Defaults to -Inf and Inf (full panel). Use I() for normalised coordinates (npc). Note: filtering only works when both the crop bound and the intercept are in data coordinates. npc bounds affect extent only.
colour	Inherits from panel.grid.major in the set theme. May be a vector the same length as the total number of lines.
linewidth	Inherits from panel.grid.major in the set theme. Supports rel(). May be a vector the same length as the total number of lines.
linetype	Inherits from panel.grid.major in the set theme. May be a vector the same length as the total number of lines.
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Value

A list of ggplot2 annotation layers.

See Also

axis_line(), reference_line(), panel_shade()

---

panel_shade *Annotate a shaded panel region*

Description

Draws filled rectangles over the panel. Defaults to a subtle overlay across the full panel. 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,
  layout = NULL
)
```

Arguments

xmin, xmax	Left and right edges of the rectangle in data coordinates. Defaults to -Inf and Inf. Use I() for normalised coordinates (0-1). May be a vector for multiple rectangles. Bounds may be mixed freely — e.g. xmin = I(0.5), xmax = Inf shades from 50% to the right panel edge.
ymin, ymax	Bottom and top edges of the rectangle in data coordinates. Defaults to -Inf and Inf. Use I() for normalised coordinates (0-1). May be a vector for multiple rectangles.
fill	Fill colour. Defaults to a neutral grey. May be a vector the same length as the bounds to style each rectangle individually.
alpha	Opacity. Defaults to 0.25. May be a vector.
colour	Border colour. Defaults to "transparent". May be a vector.
linewidth	Inherits from panel.border in the set theme. Supports rel(). May be a vector.
linetype	Border linetype. Defaults to 1. May be a vector.
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Value

A list containing annotation layers.

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` *Annotate a reference line*

Description

Draws one or more reference lines 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",
```

```
  arrow = NULL,
  layout = NULL
)
```

Arguments

xintercept	One or more x positions for vertical reference lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate line.
yintercept	One or more y positions for horizontal reference lines, in data coordinates or wrapped in I() for normalised panel coordinates (npc). May be a vector; each value produces a separate line.
colour	Inherits from axis.line in the set theme. May be a vector the same length as the total number of lines.
linewidth	Inherits from axis.line in the set theme. Supports rel(). May be a vector the same length as the total number of lines.
linetype	Defaults to "dashed". May be a vector the same length as the total number of lines.
arrow	A grid::arrow() specification, or a list the same length as the total number of lines. Must use list() not c() when supplying multiple values. E.g. grid::arrow(angle = 15, length = unit(1.5, "mm"), type = "closed").
layout	Controls which panels the annotation appears in. NULL (default) repeats in all panels. An integer targets a specific panel. "fixed" repeats in all panels ignoring faceting variables. See ggplot2::layer() for full details.

Details

The arrow (if any) points in the positive direction by default — rightward for xintercept lines, upward for yintercept lines.

Value

A list of ggplot2 annotation layers.

See Also

axis_line(), axis_ticks(), axis_text(), axis_bracket(), panel_shade()

---

sec_axis_spacer *Secondary axis for providing space for text annotations*

Description

A convenience wrapper around sec_axis_text() that reserves vertical (or horizontal) space above (or beside) an axis without drawing visible text. Useful for pushing axis titles away from the panel to make room for annotations added with axis_text(), axis_ticks(), or axis_bracket().

Usage

```
sec_axis_spacer(
  breaks = function(x) mean(x),
  labels = "",
  name = NULL,
  guide = guide_sec_axis_spacer(),
  ...
)
```

Arguments

breaks	A function or numeric vector giving the break position(s) used to anchor the text. Defaults to NULL, which places a single label at the midpoint of the scale — the mean of the limits for continuous scales.
labels	A character string used as the spacer. Defaults to "". Use repeated newlines (e.g. "\n") or a word to fit.
name	The name of the secondary axis. Use ggplot2::waiver() 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 guide_sec_axis_spacer(), which makes transparent ticks and lines.
...	Additional arguments passed to ggplot2::dup_axis().

Value

A ggplot2::sec_axis() object.

See Also

guide_sec_axis_spacer(), axis_text(), axis_ticks(), axis_bracket()

---

sec_axis_text *Secondary axis for text annotations*

Description

Secondary axis for text annotations

Usage

```
sec_axis_text(
  breaks = function(x) mean(x),
  labels = ggplot2::waiver(),
  name = NULL,
  guide = guide_sec_axis_text(),
  ...
)
```

Arguments

breaks	A function or numeric vector giving the break position(s) used to anchor the text. Defaults to NULL, which places a single label at the midpoint of the scale — the mean of the limits for continuous scales.
labels	One of: • 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 ggplot2::waiver() 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 guide_sec_axis_text(), which makes transparent ticks and lines.
...	Additional arguments passed to ggplot2::dup_axis().

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(), axis_text(), axis_ticks(), axis_bracket()

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"))
```

Index