Package 'ggfun'

July 3, 2026

Title Miscellaneous Functions for 'ggplot2'

Version 0.2.1

Author Guangchuang Yu [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-6485-8781>),
Shuangbin Xu [aut] (ORCID: <https://orcid.org/0000-0003-3513-5362>)

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Description Provides a collection of 'ggplot2' extensions and utilities
for creating geometric layers, applying themes, working with legends,
and modifying plot objects.

Depends R (>= 4.2.0)

Imports cli, dplyr, ggplot2, graphics, grid, rlang, scales, utils,
yulab.utils (>= 0.1.6)

Suggests ggplotify, knitr, pkgload, quarto, testthat, tidyr,
ggnewscale

VignetteBuilder quarto

ByteCompile true

License Artistic-2.0

Encoding UTF-8

URL https://github.com/YuLab-SMU/ggfun

BugReports https://github.com/YuLab-SMU/ggfun/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Repository CRAN

Date/Publication 2026-07-02 22:30:13 UTC

Contents

`element_blinds` *this element is used to control the line color of panel.grid.major/minor.x or panel.grid.major/minor.y*

Description

this element is used to control the line color of panel.grid.major/minor.x or panel.grid.major/minor.y

Usage

```
element_blinds(
  colour = c("white", "grey60"),
  axis,
  color = NULL,
  inherit.blank = FALSE
)
```

Arguments

colour	the colour of rectangular, default is c('white', 'grey60').
axis	character, require, option is y or x.
color	Color is an alias for colour
inherit.blank	Should this element inherit the existence of an element_blank among its parents? If TRUE the existence of a blank element among its parents will cause this element to be blank as well. If FALSE any blank parent element will be ignored when calculating final element state.

Examples

```
library(ggplot2)
df <- data.frame(
         x = rep(c(2, 5, 7, 9, 12), 2),
         y = rep(c(1, 2), each = 5),
         z = factor(rep(1:5, each = 2)),
         w = rep(diff(c(0, 4, 6, 8, 10, 14)), 2)
       )
ggplot(df, aes(x, y)) + geom_tile(aes(fill = z), colour = 'grey50') +
theme(panel.grid.major.y = element_blinds(color= c('white', 'grey'), axis='y'))
```

---

element_roundrect *round rectangle borders and backgrounds*

Description

round rectangle borders and backgrounds

Usage

```
element_roundrect(
  fill = NULL,
  colour = NULL,
  linewidth = NULL,
  linetype = NULL,
  color = NULL,
  r = grid::unit(0.1, "snpc"),
  inherit.blank = FALSE
)
```

Arguments

fill	Fill colour. fill_alpha() can be used to set the transparency of the fill.
colour, color	Line/border colour. Color is an alias for colour. alpha() can be used to set the transparency of the colour.
linewidth	Line/border size in mm
linetype	Line type for lines and borders respectively. An integer (0:8), a name (blank, solid, dashed, dotted, dotdash, longdash, twodash), or a string with an even number (up to eight) of hexadecimal digits which give the lengths in consecutive positions in the string.
r	the radius of the rounded corners, a unit object, default is unit(0.1, 'snpc').
inherit.blank	Should this element inherit the existence of an element_blank among its parents? If TRUE the existence of a blank element among its parents will cause this element to be blank as well. If FALSE any blank parent element will be ignored when calculating final element state.

Examples

```
library(ggplot2)
p <- ggplot(mpg, aes(displ, cty)) + geom_point()
p <- p + facet_grid(cols = vars(cyl))
p <- p + theme(strip.background=element_roundrect(fill="grey40", color=NA, r=0.15))
p
p2 <- ggplot(mtcars, aes(mpg, disp, color=factor(cyl), size=cyl)) +
      geom_point()
p2 + theme(legend.background=element_roundrect(color="#808080", linetype=2))
```

---

facet_set *facet_set*

Description

add a facet label to a ggplot or change facet label of a ggplot

Usage

```
facet_set(label, side = "t", angle = NULL)
```

Arguments

label	a character or a named vector to label the plot
side	to label the plot at which side, either 't' (top) or 'r' (right)
angle	angle of the facet label. Default is 0 for side='t' and -90 for side='r'.

Value

a ggplot with facet label

geom_cake *geom_cake*

Description

ggplot2 layer of birthday cake

Usage

```
geom_cake(mapping = NULL, data = NULL, ...)
```

Arguments

mapping	aes mapping
data	data
...	additional parameters

Value

ggplot2 layer

Author(s)

Guangchuang Yu

Examples

```
library(ggplot2)
ggplot(mtcars, aes(mpg, disp)) + geom_cake()
library(ggplot2)
ggplot(mtcars, aes(mpg, disp)) + geom_cake()
```

geom_scatter_rect *geom_scatter_rect*

Description

draw rectangle boxes as scatter points

Usage

```
geom_scatter_rect(
  mapping = NULL,
  data = NULL,
  asp = 0.6,
  width = 0.8,
  height = NULL,
  ...
)
```

Arguments

mapping	aesthetic mapping, default is NULL
data	input data, default is NULL
asp	aspect ration of rectangle box (height vs width), only works for height is missing
width	width of the rectangles, default is 0.8
height	height of the rectangles
...	additional parameters passed to 'geom_rect'

Author(s)

Guangchuang Yu

---

geom_segment_c *geom_segment_c*

Description

geom_segment_c supports coloring segment with continuous colors

Usage

```
geom_segment_c(
  mapping = NULL,
  data = NULL,
  position = "identity",
  lineend = "butt",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  arrow = NULL,
  arrow.fill = NULL,
  nsplit = 20,
  ...
)
```

Arguments

mapping	aes mapping
data	data
position	position
lineend	lineend
na.rm	logical
show.legend	logical
inherit.aes	logical
arrow	specification for arrow heads, as created by arrow().
arrow.fill	fill color to usse for the arrow head (if closed). NULL means use colour aesthetic.
nsplit	number of pieces used to split each segment for continuous colouring.
...	additional parameter

Value

add segment layer

Author(s)

Guangchuang Yu

See Also

geom_segment

Examples

```
set.seed(2019-06-28)
d = data.frame(x = rnorm(10),
               xend = rnorm(10),
               y = rnorm(10),
               yend = rnorm(10),
               v1 = rnorm(10),
               v2 = rnorm(10))
library(ggplot2)
ggplot(d) + geom_segment_c(aes(x = x, xend = xend, y=y, yend =yend, col0 = v1, col1 = v2)) +
   scale_color_viridis_c(name = "continuous colored lines") +
   theme_minimal() + theme(legend.position=c(.2, .85)) + xlab(NULL) + ylab(NULL)
```

geom_triangle *geom_triangle*

Description

ggplot2 layer of triangle

Usage

```
geom_triangle(mapping = NULL, data = NULL, ...)
```

Arguments

mapping	aes mapping
data	data
...	additional parameters

Value

ggplot2 layer

Author(s)

Shipeng Guo

Examples

```
library(ggplot2)
ggplot(mtcars, aes(mpg, disp)) + geom_triangle()
```

geom_volpoint *geom_volpoint*

Description

layer of scatter points for volcano plot to visualize differential genes

Usage

```
geom_volpoint(
  mapping = NULL,
  data = NULL,
  log2FC_cutoff = 2,
  p_cutoff = 1e-05,
  ...
)
```

Arguments

mapping	aesthetic mapping
data	input data set
log2FC_cutoff	cutoff values for log2FC
p_cutoff	cutoff values p-value or adjusted p-value
...	additional paramters passed to the layer

Value

a ggplot

---

geom_xspline *X-Spline Geometry for ggplot2*

Description

Draw an X-spline through control points with proper grouping support

Usage

```
geom_xspline(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  shape = 0,
  open = TRUE,
  rep_ends = TRUE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created.

A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).

stat — The statistical transformation to use on the data for this layer. When using a geom_*() function to construct a layer, the stat argument can be used to override the default coupling between geoms and stats. The stat argument accepts the following:

- A Stat ggproto subclass, for example StatCount.
- A string naming the stat. To give the stat as a string, strip the function name of the stat_ prefix. For example, to use stat_count(), give the stat as "count".
- For more information and other ways to specify the stat, see the layer stat documentation.

position — A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:

- The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position.
- A string naming the position adjustment. To give the position as a string, strip the function name of the position_ prefix. For example, to use position_jitter(), give the position as "jitter".
- For more information and other ways to specify the position, see the layer position documentation.

na.rm — If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

shape — A numeric vector of values between -1 and 1, which control the shape of the spline relative to the control points.

open — A logical value indicating whether the spline is an open or a closed shape.

rep_ends — For open X-splines, a logical value indicating whether the first and last control points should be replicated for drawing the curve. Ignored for closed X-splines.

show.legend — logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use TRUE. If NA, all levels are shown in legend, but unobserved levels are omitted.

inherit.aes — If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. annotation_borders().

... — Other arguments passed on to layer()'s params argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the position argument, or aesthetics that are required can *not* be passed through .... Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, colour = "red" or linewidth

= 3. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the params. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.

- When constructing a layer using a stat_*() function, the ... argument can be used to pass on parameters to the geom part of the layer. An example of this is stat_density(geom = "area", outline.type = "both"). The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a geom_*() function, the ... argument can be used to pass on parameters to the stat part of the layer. An example of this is geom_area(stat = "density", adjust = 0.5). The stat's documentation lists which parameters it can accept.
- The key_glyph argument of layer() may also be passed on through .... This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

Examples

```
library(ggplot2)

set.seed(123)
df <- data.frame(
  x = 1:10,
  y = cumsum(rnorm(10))
)

ggplot(df, aes(x, y)) +
  geom_point() +
  geom_xspline(color = "blue", linewidth = 1.2, shape=1)

ggplot(df, aes(x, y)) +
  geom_point() +
  geom_xspline(color = "blue", linewidth = 1.2, shape=-1)


df2 <- data.frame(
  x = rep(1:10, 2),
  y = c(cumsum(rnorm(10)), cumsum(rnorm(10))),
  group = rep(c("A", "B"), each = 10)
)

ggplot(df2, aes(x, y, color = group, group = group)) +
  geom_point() +
  geom_xspline(linewidth = 1) +
  scale_color_manual(values = c("A" = "tomato", "B" = "steelblue"))
```

get_aes_var *get_aes_var*

Description

extract aes mapping, compatible with ggplot2 < 2.3.0 & > 2.3.0

Usage

```
get_aes_var(mapping, var)
```

Arguments

mapping	aes mapping
var	variable

Value

mapped var

Author(s)

Guangchuang Yu

get_legend *get_legend*

Description

extract legend from a plot

Usage

```
get_legend(plot)
```

Arguments

plot	a gg or gtable object

Value

a 'gtable' object of the legend

Author(s)

Guangchuang Yu

get_plot_data *get_plot_data*

Description

extract data from a 'gg' plot

Usage

```
get_plot_data(plot, var = NULL, layer = NULL)
```

Arguments

plot	a 'gg' plot object
var	variables to be extracted
layer	specific layer to extract the data

Value

a data frame of selected variables

Author(s)

Guangchuang Yu

ggbreak2ggplot *ggbreak2ggplot*

Description

convert a ggbreak object to a ggplot object

Usage

```
ggbreak2ggplot(plot)
```

Arguments

plot	a ggbreak object

Value

a ggplot object

Author(s)

Guangchuang Yu

gglegend *gglegend*

Description

add manual setting legend

Usage

```
gglegend(mapping, data, geom, p = NULL)
```

Arguments

mapping	aes mapping for the 'geom'. The first mapping should be the one for the legend, while others maybe needed for the 'geom' (e.g., label for geom_text).
data	input data frame. If users want to mapping 'VALUE' to 'colour', the input data should contains 'VALUE' and 'colour' (actual value, e.g., 'red' and 'blue') variable.
geom	a geom to plot the data for generating the legend and the geom will be plotted invisible.
p	a ggplot object. If NULL, the 'last_plot()' will be used.

Details

add additional legend to a ggplot

Value

a ggplot object

Author(s)

Guangchuang Yu

Examples

```
library(ggplot2)
p <- ggplot(mtcars, aes(mpg, disp)) + geom_point()
data <- data.frame(colour = c("red",  "blue"), VALUE = c("A", "B"))
gglegend(aes(colour = VALUE, label=VALUE), data, geom_text, p)
```

`identify.gg` *identify*

Description

identify node by interactive click

Usage

```
## S3 method for class 'gg'
identify(x = last_plot(), col = "auto", ...)
```

Arguments

`x`	tree view
`col`	selected columns to extract. Default is "auto" which will select all columns for 'ggplot' object and 'node' column for 'ggtree' object
`...`	additional parameters, normally ignored

Value

closest data point

Author(s)

Guangchuang Yu

`is.ggbreak` *is.ggbreak*

Description

check whether a plot is a ggbreak object (including 'ggbreak', 'ggwrap' and 'ggcut' that defined in the 'ggbreak' package)

Usage

```
is.ggbreak(plot)
```

Arguments

`plot`	a plot obejct

Value

logical value

Author(s)

Guangchuang Yu

`is.ggtree` *is.ggtree*

Description

test whether input object is produced by ggtree function

Usage

```
is.ggtree(x)
```

Arguments

`x`	object

Value

TRUE or FALSE

Author(s)

Guangchuang Yu

`keybox` *keybox*

Description

draw border for each of the ggplot legends

Usage

```
keybox(p, grob = "roundrect", gp = NULL)
```

Arguments

`p`	a ggplot object
`grob`	one of 'rect' or 'roundrect'
`gp`	graphic parameter

Value

grob object

Author(s)

Guangchuang Yu

Examples

```
library(ggplot2)
p <- ggplot(mtcars, aes(mpg, disp, color=factor(cyl), size=cyl)) + geom_point()
keybox(p, 'roundrect', gp = gpar(col = '#808080', lty = "dashed"))
```

---

set_font *set_font*

Description

setting font for ggplot (axis text, label, title, etc.)

Usage

```
set_font(p, family = "sans", fontface = NULL, size = NULL, color = NULL)
```

Arguments

p	ggplot object
family	font fammily
fontface	font face
size	font size
color	font color

Value

TableGrob object

Author(s)

Guangchuang Yu

Examples

```
library(grid)
library(ggplot2)
d <- data.frame(x=rnorm(10), y=rnorm(10), lab=LETTERS[1:10])
p <- ggplot(d, aes(x, y)) + geom_text(aes(label=lab), size=5)
set_font(p, family="Times", fontface="italic", color='firebrick')
```

set_point_legend_shape *set_point_legend_shape*

Description

override point legend set by 'aes(shape = I(shape))'

Usage

```
set_point_legend_shape(plot)
```

Arguments

plot	a 'gg' plot object

Value

an updated plot

Author(s)

Guangchuang Yu

stat_hist *Histogram for continuous data*

Description

The output of geom_histogram() is mostly different from what produced by hist(). The geom_hist() is designed to use mid point breaks and counts calculated by hist, and thus produce identical figure with hist().

Usage

```
stat_hist(
  mapping = NULL,
  data = NULL,
  geom = "bar",
  position = "identity",
  breaks = "Sturges",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

```
geom_hist(
  mapping = NULL,
  data = NULL,
  stat = "hist",
  position = "identity",
  breaks = "Sturges",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
geom, stat	Override the default connection between geom_bar() and stat_count(). For more information about overriding these connections, see how the stat and geom arguments work.
position	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following: • The result of calling a position function, such as position_jitter(). This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the position_ prefix. For example, to use position_jitter(), give the position as "jitter". • For more information and other ways to specify the position, see the layer position documentation.
breaks	the number of cells or an algorithm to compute it, see also hist().
...	Other arguments passed on to layer()'s params argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the position argument, or aesthetics that are required can *not* be passed through .... Unknown arguments that are not part of the 4 categories below are ignored.

- Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, `colour = "red"` or `linewidth = 3`. The geom's documentation has an **Aesthetics** section that lists the available options. The 'required' aesthetics cannot be passed on to the `params`. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.
- When constructing a layer using a `stat_*()` function, the `...` argument can be used to pass on parameters to the `geom` part of the layer. An example of this is `stat_density(geom = "area", outline.type = "both")`. The geom's documentation lists which parameters it can accept.
- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The stat's documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `...`. This can be one of the functions described as key glyphs, to change the display of the layer in the legend.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

`show.legend` logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.

`inherit.aes` If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.

Value

ggplot layer

Examples

```
set.seed(42)
x <- rnorm(100)
dd = data.frame(x = c(x, x+2), type=rep(LETTERS[1:2], each=100))

require(ggplot2)
ggplot(dd, aes(x)) + geom_hist(breaks=15, color="steelblue") +
    geom_text(stat="hist", breaks=10, vjust=-1, size=5)

ggplot(dd, aes(x)) + geom_hist(breaks=15, aes(fill=type), position='dodge')
```

td_filter *td-filter*

Description

filter data for tree annotation layer

Usage

```
td_filter(..., .f = NULL)
```

Arguments

...	Expressions that return a logical value.
.f	a function (if any, defaults to NULL) that pre-operate the data

Details

The 'td_filter()' function returns another function that can be used to subset ggtree() plot data. The function can be passed to the 'data' parameter of geom layer to perform subsetting. All rows that satisy your conditions will be retained.

Value

A function to filter ggtree plot data using conditions defined by '...'.

Author(s)

Guangchuang Yu

References

For more detailed demonstration of this function, please refer to chapter 12.5.1 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

See Also

filter

Examples

```
## Not run:
tree <- rtree(30)
## similar to 'ggtree(tree) + geom_tippoint()'
ggtree(tree) + geom_point(data = td_filter(isTip))

## End(Not run)
```

td_mutate *td-mutate*

Description

mutate data for tree annotation layer

Usage

```
td_mutate(..., .f = NULL)
```

Arguments

...	additional parameters that pass to dplyr::mutate
.f	a function (if any, defaults to NULL) that pre-operate the data

Details

The 'td_mutate()' function returns another function that can be used to mutate ggtree() plot data. The function can be passed to the 'data' parameter of geom layer to perform adding new variables and preserving existing ones.

Value

A function to mutate ggtree plot data

See Also

mutate

td_unnest *td-unnest*

Description

flatterns a list-column of data frame

Usage

```
td_unnest(cols, ..., .f = NULL)
```

Arguments

cols	columns to unnest
...	additional parameters that pass to tidyr::unnest
.f	a function (if any, defaults to NULL) that pre-operate the data

Details

The 'td_unnest' function returns another function that can be used to unnest ggtree() plot data. The function can be passed to the 'data' parameter of a geom layer to flattern list-cloumn tree data.

Value

A function to unnest ggtree plot data

Author(s)

Guangchuang Yu

References

For demonstration of this function, please refer to chapter 12.5.2 of *Data Integration, Manipulation and Visualization of Phylogenetic Trees* http://yulab-smu.top/treedata-book/index.html by Guangchuang Yu.

See Also

unnest

---

theme_blinds *the theme of blind-like*

Description

the theme of blind-like

Usage

```
theme_blinds(colour = c("white", "grey"), axis = "y", ...)
```

Arguments

colour	the colour of rectangular, default is c('white', 'grey60').
axis	character which grid of axis will be filled, default is 'y'.
...	additional parameters that passed to theme function.

Value

ggplot2 theme

Examples

```
library(ggplot2)
iris |> tidyr::pivot_longer(
    cols = !Species,
    names_to = 'var',
    values_to = 'value'
  ) |>
ggplot(
  aes(x=var, y=Species, color=value, size=value)
) +
geom_point() -> p
p +
theme_blinds(
  colour = c('grey90', 'white'),
  axis = 'y',
  axis.line.y=element_line()
)
p +
theme_blinds(
  colour = c('grey90', 'white'),
  axis = 'x',
  axis.line.x = element_line()
)
```

---

theme_fp *theme_fp*

Description

theme format painter

Usage

```
theme_fp(x, i)
```

Arguments

x	ggplot object to provide theme format
i	the element of a theme provided by x

Details

It applies theme element (i) from a ggplot (x) to another ggplot object

Value

theme element

Author(s)

Guangchuang Yu and Shuangbin Xu

`theme_nothing` *theme_nothing*

Description

A theme that only show the plot panel

Usage

```
theme_nothing(base_size = 11, base_family = "")
```

Arguments

`base_size`	font size
`base_family`	font family

Value

ggplot2 theme

Author(s)

Guangchuang Yu

`theme_noxaxis` *theme_noxaxis*

Description

A theme that only show y-axis

Usage

```
theme_noxaxis(color = "black", ...)

theme_noyaxis(color = "black", ...)

theme_noaxis(...)
```

Arguments

`color`	color of y-axis
`...`	additional parameters that passed to theme()

Value

ggplot2 theme

Author(s)

Guangchuang Yu

---

theme_no_margin *theme_no_margin*

Description

A theme that has no margin

Usage

```
theme_no_margin(...)
```

Arguments

...	additional parameters that passed to theme()

Value

ggplot2 theme

Author(s)

Guangchuang Yu

---

theme_stamp *the theme of blind-like alias of theme_blinds*

Description

the theme of blind-like alias of theme_blinds

Usage

```
theme_stamp(colour = c("white", "grey"), axis = "y", ...)
```

Arguments

colour	the colour of rectangular, default is c('white', 'grey60').
axis	character which grid of axis will be filled, default is 'y'.
...	additional parameters that passed to theme function.

theme_transparent *theme_transparent*

Description

transparent background theme

Usage

```
theme_transparent(...)
```

Arguments

...	additional parameter to tweak the theme

Value

ggplot object

Author(s)

Guangchuang Yu with contributions from Hugo Gruson

volplot *volplot*

Description

volcano plot

Usage

```
volplot(data, mapping, log2FC_cutoff = 2, p_cutoff = 1e-05, ...)
```

Arguments

data	input data set
mapping	aesthetic mapping
log2FC_cutoff	cutoff values for log2FC
p_cutoff	cutoff values p-value or adjusted p-value
...	additional paramters passed to the 'geom_volpoint' layer

Value

a ggplot

yrange *plot range of a ggplot object*

Description

extract x or y ranges of a ggplot

Usage

```
yrange(gg, type = "limit", region = "panel")

xrange(gg, type = "limit", region = "panel")

ggrange(gg, var, type = "limit", region = "panel")
```

Arguments

gg	a ggplot object
type	one of 'limit' or 'range', if 'region == "plot"', to extract plot limit or plot data range
region	one of 'panel' or 'plot' to indicate extracting range based on the plot panel (scale expand will be counted) or plot data (scale expand will not be counted)
var	either 'x' or 'y'

Value

range of selected axis

Author(s)

Guangchuang Yu

%<+% *%<+%*

Description

This operator attaches annotation data to a ggtree or ggsc graphic object

Usage

```
p %<+% data
```

Arguments

p	ggplot2 object, such as ggtree or ggsc graphic object.
data	data.frame, which must contains a column of node, or the first column of taxa labels, when p is a ggtree object. Or it must contains columns of .BarcodeID, when p is a ggsc object and p$data does not contain a column of features, if it contains, the data must also contains a column of features.

Value

ggplot object with annotation data added

Index