

# Package ‘ggfun’

January 19, 2024

**Title** Miscellaneous Functions for 'ggplot2'

**Version** 0.1.4

**Description** Useful functions and utilities for 'ggplot' object (e.g., geometric layers, themes, and utilities to edit the object).

**Depends** R (>= 4.1.0)

**Imports** cli, ggplot2, grid, rlang, utils

**Suggests** ggplotify, knitr, rmarkdown, prettydoc, tidy, ggnewscale

**VignetteBuilder** knitr

**ByteCompile** true

**License** Artistic-2.0

**Encoding** UTF-8

**URL** <https://github.com/YuLab-SMU/ggfun>

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

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Guangchuang Yu [aut, cre, cph]

(<https://orcid.org/0000-0002-6485-8781>),

Shuangbin Xu [aut] (<https://orcid.org/0000-0003-3513-5362>)

**Maintainer** Guangchuang Yu <[guangchuangyu@gmail.com](mailto:guangchuangyu@gmail.com)>

**Repository** CRAN

**Date/Publication** 2024-01-19 05:30:06 UTC

## R topics documented:

element_blinds . . . . .	2
element_roundrect . . . . .	3
facet_set . . . . .	4
geom_cake . . . . .	4
geom_segment_c . . . . .	5

geom_triangle . . . . .	6
geom_volpoint . . . . .	7
get_aes_var . . . . .	8
get_legend . . . . .	8
get_plot_data . . . . .	9
ggbreak2ggplot . . . . .	9
gglegend . . . . .	10
identify.gg . . . . .	11
is.ggbreak . . . . .	11
is.ggtree . . . . .	12
keybox . . . . .	12
set_font . . . . .	13
set_point_legend_shape . . . . .	14
theme_blinds . . . . .	14
theme_fp . . . . .	15
theme_nothing . . . . .	16
theme_noxaxis . . . . .	16
theme_no_margin . . . . .	17
theme_stamp . . . . .	17
theme_transparent . . . . .	18
volplot . . . . .	18
yrange . . . . .	19

## Index 20

---

element_blinds	<i>this element is used to control the line color of panel.grid.major/minor.x or panel.grid.major/minor.y</i>
----------------	---

---

## 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,
  size = NULL,
  linetype = NULL,
  color = NULL,
  r = grid::unit(0.1, "snpc"),
  inherit.blank = FALSE
)
```

## Arguments

<code>fill</code>	Fill colour.
<code>colour, color</code>	Line/border colour. Color is an alias for colour.
<code>size</code>	text size in pts.
<code>linetype</code>	Line type. 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.
<code>r</code>	the radius of the rounded corners, a unit object, default is <code>unit(0.1, 'snpc')</code> .
<code>inherit.blank</code>	Should this element inherit the existence of an <code>element_blank</code> 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_segment_c	<i>geom_segment_c</i>
----------------	-----------------------

---

**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,  
  ...  
)
```

**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 use for the arrow head (if closed). NULL means use colour aesthetic.
...	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()
```

---

<code>geom_volpoint</code>	<code>geom_volpoint</code>
----------------------------	----------------------------

---

**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

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	<i>identify</i>
-------------	-----------------

---

**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	<i>is.ggbreak</i>
------------	-------------------

---

**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 object
------	---------------

**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**

<code>p</code>	ggplot object
<code>family</code>	font fammily
<code>fontface</code>	font face
<code>size</code>	font size
<code>color</code>	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

---

```
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	<i>theme_nothing</i>
---------------	----------------------

---

**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	<i>theme_noxaxis</i>
---------------	----------------------

---

**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	<i>theme_no_margin</i>
-----------------	------------------------

---

**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	<i>the theme of blind-like alias of theme_blinds</i>
-------------	--

---

**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	<i>theme_transparent</i>
-------------------	--------------------------

---

**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	<i>volplot</i>
---------	----------------

---

**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	<i>plot range of a ggplot object</i>
--------	--------------------------------------

---

**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

# Index

element\_blinds, 2  
element\_roundrect, 3

facet\_set, 4

geom\_cake, 4  
geom\_segment, 6  
geom\_segment\_c, 5  
geom\_triangle, 6  
geom\_volpoint, 7  
get\_aes\_var, 8  
get\_legend, 8  
get\_plot\_data, 9  
ggbreak2ggplot, 9  
gglegend, 10  
ggrange (yrange), 19

identify.gg, 11  
is.ggbreak, 11  
is.ggtree, 12

keybox, 12

set\_font, 13  
set\_point\_legend\_shape, 14

theme\_blinds, 14  
theme\_fp, 15  
theme\_no\_margin, 17  
theme\_noaxis (theme\_noxaxis), 16  
theme\_nothing, 16  
theme\_noxaxis, 16  
theme\_noyaxis (theme\_noxaxis), 16  
theme\_stamp, 17  
theme\_transparent, 18

volplot, 18

xrange (yrange), 19

yrange, 19