Package 'forestploter'

September 30, 2026

Title Create a Flexible Forest Plot

Version 1.2.0

Description Create a forest plot based on the layout of the data. Confidence intervals in multiple columns by groups can be done easily. The plot is built step by step with the pipe, adding the axis, the labels and a style, editing the plot, inserting/adding text, and much more.

License MIT + file LICENSE

Depends R (>= 4.1.0)

URL https://github.com/adayim/forestploter, https://adayim.github.io/forestploter/

BugReports https://github.com/adayim/forestploter/issues

Encoding UTF-8

Imports grid, gridExtra, gtable

Suggests gridmicrotex, rmarkdown, knitr, vdiffr, testthat (>= 3.0.0), covr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation no

Author Alimu Dayimu [aut, cre] (ORCID: <https://orcid.org/0000-0001-9998-7463>)

Maintainer Alimu Dayimu <ad938@cam.ac.uk>

Repository CRAN

Date/Publication 2026-09-30 17:10:07 UTC

Contents

add_border . . . 2
add_grob . . . 3
add_text . . . 4
edit_plot . . . 5
forest . . . 6
forest_style . . . 9
forest_theme . . . 12
get_wh . . . 16
insert_text . . . 17
makeci . . . 18
make_boxplot . . . 19
make_summary . . . 21
print.forestplot . . . 22
print.forest_style . . . 23
scale_sizes . . . 23
set_labs . . . 25
set_style . . . 26
set_xaxis . . . 27

Index **30**

`add_border` *Add border to cells*

Description

Add border to any cells at any side.

Usage

```
add_border(
  plot,
  row = NULL,
  col = NULL,
  part = c("body", "header"),
  where = c("bottom", "left", "top", "right"),
  gp = gpar(lwd = 2)
)
```

Arguments

`plot`	A forest plot object.
`row`	A numeric value or vector indicating row number to add border. This is corresponding to the data row number. Remember to account for any text inserted. A border will be drawn to all rows if this is omitted.

col	A numeric value or vector indicating the columns to add border. A border will be drawn to all columns if this is omitted.
part	The border will be added to "body" (default) or "header".
where	Where to draw the border of the cell, possible values are "bottom" (default), "left", "top" and "right"
gp	An object of class "gpar", graphical parameter to be passed to segmentsGrob.

Value

A gtable object.

See Also

gpar segmentsGrob gtable_add_grob

---

add_grob *Add grob in cells*

Description

Draw grobs in any cells.

Usage

```
add_grob(
  plot,
  row = NULL,
  col = NULL,
  part = c("body", "header"),
  order = c("top", "text", "background", "bottom"),
  gb_fn,
  ...
)
```

Arguments

plot	A forest plot object.
row	A numeric value or vector indicating row(s) to draw a grob.
col	A numeric value or vector indicating the columns to draw a grob.
part	The grob will be added to "body" (default) or "header".
order	Order in which the grobs should be plotted. Use 'top' (default) to draw the grob above everything, 'text' on the top of text given by plot data but below everything else, 'background' plot on the top of background but below everything else, 'bottom' below everything.
gb_fn	Grob function
...	Other parameters to be passed to gb_fn.

Value

A gtable object.

See Also

gtable_add_grob

---

add_text *Add text to forest plot*

Description

This function can be used to add text to a forest plot. The text can span multiple rows and columns. The height of the row will be adjusted accordingly if the text is added to only one row. The width of the text may exceed the columns provided if the text is too long.

Usage

```
add_text(
  plot,
  text,
  row = NULL,
  col = NULL,
  part = c("body", "header"),
  just = c("center", "left", "right"),
  gp = gpar(),
  padding = unit(1, "mm"),
  parse = FALSE
)
```

Arguments

plot	A forest plot object.
text	A character or expression vector, see textGrob.
row	Row to add the text, this will be ignored if the part is "header".
col	A numeric value or vector indicating the columns the text will be added. The text will span over the column if a vector is given.
part	Part to add text, "body" (default) or "header".
just	The justification of the text, "center" (default), "left" or "right".
gp	An object of class "gpar", this is the graphical parameter settings of the text. See gpar.
padding	Padding of the text, default is unit(1, "mm")
parse	Logical, behaviour for parsing text as plotmath, see plotmath

Value

A gtable object.

See Also

gtable gpar textGrob gtable_add_grob

`edit_plot` *Edit forest plot*

Description

This function is used to edit the graphical parameters of text and background of the forest plot.

Usage

```
edit_plot(
  plot,
  row = NULL,
  col = NULL,
  part = c("body", "header"),
  which = c("text", "background", "ci"),
  gp = gpar(),
  ...
)
```

Arguments

`plot`	A forest plot object.
`row`	A numeric value or vector indicating row number to edit in the dataset. Will edit the whole row if left blank for the body. This will be ignored if the `part` is "header".
`col`	A numeric value or vector indicating column to edit in the dataset. Will edit the whole column if left blank.
`part`	Part to edit, `"body"` (default) or `"header"`.
`which`	Which element to edit, `"text"`, `"background"` or `"ci"` (confidence interval). This will not edit diamond shaped summary CI, please change it with `forest_theme`. Also, change in `ci` will not have any impact on the legend.
`gp`	Pass `gpar` parameters, see `gpar`. It should be passed as `gpar(col = "red")`. For `which = "ci"`, please refer to `forest_theme` `ci_*` parameters for the editable elements.
`...`	Other parameters to be passed to the grobs. See `textGrob` for the `"text"` part and `rectGrob` for `"background"`. This is ignored when `which = "ci"` because non-graphical parameters cannot be changed for the confidence interval.

Value

A gtable object.

See Also

gpar editGrob forest_theme textGrob rectGrob

---

forest *Forest plot*

Description

A data frame will be used for the basic layout of the forest plot. Graphical parameters can be set using the forest_style function.

forest draws the table and the confidence intervals. The other parts of the plot are added with functions that take the plot as their first argument, so they can be chained with the pipe |>:

- set_xaxis Limits, tick marks and scale of the x-axis, and vertical lines
- set_labs Title, x-axis labels, footnote, arrow labels and legend labels
- scale_sizes Point sizes scaled by study weights
- set_style Graphical parameters

These functions build the plot again, so they must be used before the plot is edited with edit_plot, add_text, insert_text, add_border or add_grob. The plot stays a gtable at every step, and can be combined with other plots, e.g. with patchwork::wrap_elements.

Usage

```
forest(
  data,
  est,
  lower,
  upper,
  sizes = 0.4,
  ref_line = NULL,
  ci_column,
  is_summary = NULL,
  nudge_y = 0,
  fn_ci = makeci,
  fn_summary = make_summary,
  index_args = NULL,
  style = NULL,
  ...
)
```

Arguments

data	Data to be displayed in the forest plot
est	Point estimation. Can be a list for multiple columns and/or multiple groups. If the length of the list is larger than then length of ci_column, then the values reused for each column and considered as different groups.
lower	Lower bound of the confidence interval, same as est.
upper	Upper bound of the confidence interval, same as est.
sizes	Size of the point estimation box, can be a vector or a list. The value is a multiple of one line of text, so 1 draws a point as tall as the base_size of the theme. The same scale applies to the summary diamond. Values are used as they are, unless scale_sizes is used to read them as study weights; useful values are roughly between 0.2 and 1.5, and a warning is given when the plot is drawn if they are outside 0.1 to 2.
ref_line	X-axis coordinates of the reference line, the value of no effect. If NULL (default), it is 1 if the x-axis is on a log scale (see set_xaxis) and 0 otherwise. Provide an atomic vector if different reference line for each ci_column is desired.
ci_column	Column number of the data the CI will be displayed.
is_summary	A logical vector indicating if the value is a summary value, which will have a diamond shape for the estimate. With multiple groups the diamonds are stacked in the same cell and the summary rows are made taller to fit them, so a larger nudge_y may be wanted.
nudge_y	Vertical adjustment to nudge groups by, must be within 0 to 1. Defaults to 0; for grouped forest plots a value of 0 is bumped to 0.1 automatically so that group CIs do not overplot. Set explicitly to override.
fn_ci	Name of the function to draw confidence interval, default is makeci. You can specify your own drawing function to draw the confidence interval, but the function needs to accept arguments "est", "lower", "upper", "sizes", "xlim", "pch", "gp", "t_height", "nudge_y". Please refer to the makeci function for the details of these parameters.
fn_summary	Name of the function to draw summary confidence interval, default is make_summary. You can specify your own drawing function to draw the summary confidence interval, but the function needs to accept arguments "est", "lower", "upper", "sizes", "xlim", "gp". Please refer to the make_summary function for the details of these parameters.
index_args	A character vector, name of the arguments used for indexing the row and column. This should be the name of the arguments that is working the same way as est, lower and upper. Check out the examples in the make_boxplot.
style	Style of the forest plot created with forest_style. A theme created with the superseded forest_theme is also accepted. The style can also be set or changed later with set_style.
...	Other arguments passed on to the fn_ci and fn_summary, or named in index_args. An argument none of them takes gives an error, as it would not be used. The arguments of earlier versions are also accepted here, with a message the first time each of them is used in a session: use set_xaxis instead of xlim, ticks_at, ticks_digits, ticks_minor, x_trans and vert_line, set_labs instead of arrow_lab, xlab, title and footnote, and style instead of theme.

Value

A forest plot object, a gtable of class forestplot.

See Also

gtable tableGrob forest_style set_xaxis set_labs scale_sizes set_style make_boxplot makeci make_summary

Examples

```
library(grid)
# Read provided sample example data
dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))

# Keep needed columns
dt <- dt[,1:6]

# indent the subgroup if there is a number in the placebo column
dt$Subgroup <- ifelse(is.na(dt$Placebo),
                      dt$Subgroup,
                      paste0("   ", dt$Subgroup))

# NA to blank or NA will be transformed to carachter.
dt$Treatment <- ifelse(is.na(dt$Treatment), "", dt$Treatment)
dt$Placebo <- ifelse(is.na(dt$Placebo), "", dt$Placebo)
dt$se <- (log(dt$hi) - log(dt$est))/1.96

# Add blank column for the forest plot to display CI.
# Adjust the column width with space.
dt$` ` <- paste(rep(" ", 20), collapse = " ")

# Create confidence interval column to display
dt$`HR (95% CI)` <- ifelse(is.na(dt$se), "",
                           sprintf("%.2f (%.2f to %.2f)",
                                   dt$est, dt$low, dt$hi))

# Define a style
st <- forest_style(base_size = 10,
                   ref_line = gpar(col = "red"),
                   footnote = gpar(col = "#636363", fontface = "italic"))

# Draw the plot and add the axis and labels with a pipe
p <- forest(dt[,c(1:3, 8:9)],
            est = dt$est,
            lower = dt$low,
            upper = dt$hi,
            sizes = dt$se,
            ci_column = 4,
            ref_line = 1,
            style = st) |>
  set_xaxis(xlim = c(0, 4), ticks_at = c(0.5, 1, 2, 3)) |>
  set_labs(arrow = c("Placebo Better", "Treatment Better"),
```

```
        footnote = "This is the demo data. Please feel free to change\nanything you want.")

# Print plot
plot(p)
```

`forest_style` *Forest plot style*

Description

Set the look of a forest plot. Each part of the plot takes a `gpar` object, and only the settings given are changed, everything else keeps its default. A style can be passed to the `theme` argument of `forest` or applied to a plot with `set_style`, so the same style can be reused for many plots. To change some settings of a plot and keep the rest, give them to `set_style` instead.

The text itself, such as the title or the legend labels, is set with `set_labs`.

Usage

```
forest_style(
  base_size = 12,
  base_family = "",
  parse = NULL,
  ci = gpar(),
  ci_pch = 15,
  ci_t_height = NULL,
  summary = gpar(),
  ref_line = gpar(),
  vline = gpar(),
  xaxis = gpar(),
  xlab = gpar(),
  xlab_adjust = c("refline", "center"),
  title = gpar(),
  title_just = c("left", "right", "center"),
  footnote = gpar(),
  arrow = gpar(),
  arrow_type = c("open", "closed"),
  arrow_length = 0.05,
  arrow_label_just = c("start", "end"),
  legend = gpar(),
  legend_position = c("right", "top", "bottom", "none"),
  legend_ncol = 1,
  legend_byrow = TRUE,
  body = gpar(),
  header = gpar(),
  fit = c("none", "width", "both"),
  ...
)
```

Arguments

base_size	The size of text.
base_family	The font family, the font of the device by default.
parse	Whether text is read as plotmath expressions, see plotmath. This applies to the title, x-axis labels, footnote, arrow labels and legend labels; text in the table cells is parsed with parse in core and colhead through .... By default only the footnote is parsed, TRUE parses all of them where the text is a valid expression and FALSE parses none.
ci	Confidence intervals, col, fill, lty, lwd and alpha are used. Provide a vector for each group of a grouped forest plot. fill is only used if ci_pch is within 15:25 and alpha must be a single value; a small vertical line marks the point estimate if it is not 1.
ci_pch	Shape of the point estimation, reused for each group if a single value is given.
ci_t_height	The height of the T end of the confidence intervals. No T end is drawn by default.
summary	Diamond shaped summary confidence intervals, col and fill are used.
ref_line	Reference line, its position is set with ref_line of forest.
vline	Vertical lines, their positions are set with vline of set_xaxis. lwd, lty and col can be vectors with one value for each line.
xaxis	X-axis line, tick marks and tick labels.
xlab	X-axis labels.
xlab_adjust	Align the x-axis labels to the reference line "refline" (default) or to the center of the x-axis "center".
title	Title.
title_just	The justification of the title, "left" (default), "right" or "center".
footnote	Footnote.
arrow	Arrows and their labels.
arrow_type	Type of the arrow head, "open" (default) or "closed", see arrow.
arrow_length	The length of the arrow head, a unit or a number in inches. The default is 0.05 inches.
arrow_label_just	Align the arrow labels to the starting point of the arrows "start" (default) or to their ending point "end".
legend	Legend text.
legend_position	Position of the legend, "right" (default), "top", "bottom" or "none" to hide the legend.
legend_ncol	The number of columns of the legend, see legendGrob.
legend_byrow	Whether the rows of the legend are filled first, see legendGrob.
body	Text and background of the body of the table, a short form of core in ...: col, fontsize, fontface, fontfamily, cex, lineheight and alpha are used for the text, fill for the background. A vector is recycled over the rows.

header — Text and background of the header of the table, a short form of colhead in ..., same as body.

fit — How the plot uses the space it is drawn in, for example the size given to ggplot2::ggsave or a panel of patchwork. With "none" (default) the plot keeps its natural size, the size given by get_wh, and is centred in the space. With "width" the CI columns take the free width, in proportion to their natural width, and become narrower when space is short. "both" also shares the free height between the rows of the table. Text always keeps its size.

... — Settings passed on to the theme of the table, see tableGrob: core for the body of the table and colhead for its header, each a list with fg_params for the text and bg_params for the background. For example core = list(fg_params = list(hjust = 1, x = 0.9)) aligns the text of the body to the right. body and header above are applied on top of them, so settings given in both places come from body and header. The border of a cell takes the colour of its fill, so that there is no gap between the cells, unless bg_params of core or colhead gives it a colour.

Value

A forest_style object.

See Also

set_style forest set_labs gpar tableGrob

Examples

```
library(grid)
# Read provided sample example data
dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))
dt <- dt[1:8, ]

# NA to blank or NA will be transformed to character
dt$Treatment <- ifelse(is.na(dt$Treatment), "", dt$Treatment)
dt$Placebo <- ifelse(is.na(dt$Placebo), "", dt$Placebo)

# Add blank columns for the forest plot to display CI.
# Adjust the column width with space.
dt$` ` <- paste(rep(" ", 20), collapse = " ")
dt$`  ` <- paste(rep(" ", 20), collapse = " ")

# A style that can be reused for other plots
st <- forest_style(base_size = 10,
                   ref_line = gpar(col = "red"),
                   vline = gpar(col = "grey60"),
                   footnote = gpar(col = "#636363", fontface = "italic"),
                   arrow_type = "closed",
                   title_just = "center")

# Add the axis and the text to the plot
p <- forest(dt[, c(1:3, 19)],
```

```
            est = dt$est,
            lower = dt$low,
            upper = dt$hi,
            sizes = dt$est,
            ci_column = 4,
            ref_line = 1,
            style = st) |>
  set_xaxis(x_trans = "log",
            xlim = c(0.25, 4),
            ticks_at = c(0.5, 1, 2, 4),
            vline = c(0.5, 2)) |>
  set_labs(title = "Subgroup analysis",
           xlab = "Hazard ratio",
           arrow = c("Placebo Better", "Treatment Better"),
           footnote = "This is the demo data.") |>
  scale_sizes(method = "range", range = c(0.3, 0.8))

plot(p)

# Change the style, let the CI column take the free width of the page and
# remove the footnote
p <- p |>
  set_style(base_size = 12, title = gpar(col = "blue"), fit = "width") |>
  set_labs(footnote = NULL)

plot(p)

# Grouped CIs in two columns, with a legend
p <- forest(dt[, c(1, 19, 20)],
            est = list(dt$est_gp1, dt$est_gp2, dt$est_gp3, dt$est_gp4),
            lower = list(dt$low_gp1, dt$low_gp2, dt$low_gp3, dt$low_gp4),
            upper = list(dt$hi_gp1, dt$hi_gp2, dt$hi_gp3, dt$hi_gp4),
            ci_column = c(2, 3),
            ref_line = 1,
            nudge_y = 0.2,
            style = forest_style(ci = gpar(col = c("#377eb8", "#4daf4a")),
                                 legend_position = "bottom")) |>
  set_xaxis(x_trans = "log", xlim = list(c(0.1, 5), NA)) |>
  set_labs(xlab = c("CVD outcome", "COPD outcome"),
           legend_title = "Group",
           legend_labels = c("Trt 1", "Trt 2"))

plot(p)
```

`forest_theme` *Forest plot default theme*

Description

Default theme for the forest plot. Other parameters can also be passed and will be forwarded to the corresponding elements of the forest plot.

forest_theme is superseded by forest_style, see the section below for how its arguments map onto the new functions.

- ci_* Control the graphical parameters of confidence intervals
- legend_* Control the graphical parameters of legend
- xaxis_* Control the graphical parameters of x-axis
- refline_* Control the graphical parameters of reference line
- vertline_* Control the graphical parameters of vertical line
- summary_* Control the graphical parameters of diamond shaped summary CI
- footnote_* Control the graphical parameters of footnote
- title_* Control the graphical parameters of title
- arrow_* Control the graphical parameters of arrow

See gpar for more details.

Usage

```
forest_theme(
  base_size = 12,
  base_family = "",
  ci_pch = 15,
  ci_col = "black",
  ci_alpha = 1,
  ci_fill = NULL,
  ci_lty = 1,
  ci_lwd = 1,
  ci_Theight = NULL,
  legend_name = "Group",
  legend_position = "right",
  legend_value = "",
  legend_gp = gpar(),
  legend_ncol = 1,
  legend_byrow = TRUE,
  xaxis_gp = gpar(),
  refline_gp = gpar(),
  vertline_lwd = 1,
  vertline_lty = "dashed",
  vertline_col = "grey20",
  summary_col = "#4575b4",
  summary_fill = summary_col,
  footnote_gp = gpar(),
  footnote_parse = TRUE,
  title_just = c("left", "right", "center"),
  title_gp = gpar(),
  arrow_type = c("open", "closed"),
  arrow_label_just = c("start", "end"),
  arrow_length = 0.05,
```

```
  arrow_gp = gpar(),
  xlab_adjust = c("refline", "center"),
  xlab_gp = gpar(),
  ...
)
```

Arguments

base_size	The size of text
base_family	The font family
ci_pch	Shape of the point estimation. It will be reused if the forest plot is grouped.
ci_col	Color of the CI. A vector of colors should be provided for a grouped forest plot. An internal color set will be used if not provided.
ci_alpha	Scalar value, alpha channel for transparency of the point estimate. A small vertical line will be added to mark the location of the point estimate if this is not equal to 1.
ci_fill	Fill color of the point estimation. A vector of colors should be provided for a grouped forest plot. If this is NULL (default), the value will be inherited from ci_col. This is only effective if ci_pch is within 15:25.
ci_lty	Line type of the CI. A vector of line types should be provided for a grouped forest plot.
ci_lwd	Line width of the CI. A vector of line widths should be provided for a grouped forest plot.
ci_Theight	A unit specifying the height of the T end of the CI. If set to NULL (default), no T end will be drawn.
legend_name	Title of the legend.
legend_position	Position of the legend, "right", "top", "bottom" or "none" to suppress the legend.
legend_value	Legend labels (expressions). A vector should be provided for a grouped forest plot. Defaults to "Group 1", "Group 2", ... if not provided.
legend_gp	gpar graphical parameters of legend, see gpar.
legend_ncol	integer; the number of columns, see legendGrob.
legend_byrow	logical indicating whether rows of the legend are filled first, see legendGrob.
xaxis_gp	gpar graphical parameters of x-axis, see gpar.
refline_gp	gpar graphical parameters of reference line, see gpar.
vertline_lwd	Line width for extra vertical line. A vector can be provided for each vertical line, and the values will be recycled if not enough values are given.
vertline_lty	Line type for extra vertical line. Works same as vertline_lwd.
vertline_col	Line color for the extra vertical line. Works same as vertline_lwd.
summary_col	Color for borders of the summary diamond shape.
summary_fill	Color for filling the summary diamond shape.

footnote_gp	gpar graphical parameters of footnote, see gpar.
footnote_parse	Parse footnote text (default).
title_just	The justification of the title, default is 'left'.
title_gp	gpar graphical parameters of title, see gpar.
arrow_type	Type of the arrow below x-axis, see arrow.
arrow_label_just	The justification of the arrow label relative to arrow. Control the arrow label to align to the starting point of the arrow "start" (default) or the ending point of the arrow "end".
arrow_length	The length of the arrow head, default is 0.05. See arrow.
arrow_gp	gpar graphical parameters of arrow, see gpar.
xlab_adjust	Control the alignment of xlab to reference line (default) or center of the x-axis.
xlab_gp	gpar graphical parameters of xlab, see gpar.
...	Other parameters passed to table. See tableGrob for details.

Value

A list.

Moving to forest_style()

forest_style takes one gpar for each part of the plot, and the text of the legend is set with set_labs. Themes created with forest_theme still work: give them to the style argument of forest or to set_style.

- base_size, base_family, ci_pch, xlab_adjust, title_just, arrow_type, arrow_length, arrow_label_just, legend_position, legend_ncol, legend_byrow: the same arguments of forest_style.
- ci_col, ci_fill, ci_lty, ci_lwd, ci_alpha: forest_style(ci = gpar(col, fill, lty, lwd, alpha)).
- ci_Theight: forest_style(ci_t_height).
- summary_col, summary_fill: forest_style(summary = gpar(col, fill)).
- vertline_lwd, vertline_lty, vertline_col: forest_style(vline = gpar(lwd, lty, col)).
- refline_gp: forest_style(ref_line).
- xaxis_gp, xlab_gp, title_gp, footnote_gp, arrow_gp, legend_gp: the arguments of forest_style without _gp, e.g. forest_style(title = gpar(col = "red")).
- footnote_parse: forest_style(parse), which also applies to the title, x-axis labels, arrow labels and legend labels.
- legend_name, legend_value: legend_title and legend_labels of set_labs.
- The fill of core and colhead, e.g. core = list(bg_params = list(fill = "white")): forest_style(body = gpar(fill = "white")) and forest_style(header = gpar(fill = "white")).
- Other table settings in ...: passed to ... of forest_style in the same way.

See Also

tableGrob forest textGrob gpar arrow segmentsGrob linesGrob pointsGrob legendGrob

get_wh *Get width and height of the forestplot*

Description

get_wh can be used to find the correct width and height of the forestplot for saving, as the width and height are difficult to estimate otherwise.

Usage

```
get_wh(plot, unit = c("in", "cm", "mm"))
```

Arguments

plot	A forest plot object.
unit	Unit for the returned width and height. One of "in", "cm", or "mm".

Details

This is the natural size of the plot, where every column and row fits its content. By default a plot drawn in a larger space keeps this size and is centred. With fit of forest_style the CI columns, and the rows if asked for, take the space instead.

Value

A named vector of width and height

Examples

```
## Not run:
 dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))
 dt <- dt[1:6,1:6]

 dt$` ` <- paste(rep(" ", 20), collapse = " ")

 p <- forest(dt[,c(1:3, 7)],
             est = dt$est,
             lower = dt$low,
             upper = dt$hi,
             ci_column = 4)

# get_wh example
p_wh <- get_wh(p)
pdf('test.pdf',width = p_wh[1], height = p_wh[2])
plot(p)
dev.off()

## End(Not run)
```

`insert_text` *Insert text to forest plot*

Description

This function can be used to insert text into a forest plot. Remember to adjust for the row number if you have added text before, including the header. This is achieved by inserting new row(s) into the plot and will affect subsequent row numbers. A text vector can be inserted into multiple columns or rows.

Usage

```
insert_text(
  plot,
  text,
  row = NULL,
  col = NULL,
  part = c("body", "header"),
  just = c("center", "left", "right"),
  before = TRUE,
  gp = gpar(),
  padding = unit(1, "mm"),
  parse = FALSE
)
```

Arguments

`plot`	A forest plot object.
`text`	A character or expression vector, see `textGrob`.
`row`	Row to insert the text, this will be ignored if the `part` is "header".
`col`	A numeric value or vector indicating the columns the text will be added. The text will span over the column if a vector is given.
`part`	Part to insert text, `"body"` (default) or `"header"`.
`just`	The justification of the text, `"center"` (default), `"left"` or `"right"`.
`before`	Indicating the text will be inserted before or after the row.
`gp`	An object of class `"gpar"`, this is the graphical parameter settings of the text. See `gpar`.
`padding`	Padding of the text, default is `unit(1, "mm")`
`parse`	Logical, behaviour for parsing text as plotmath, see `plotmath`

Value

A `gtable` object.

See Also

gpar textGrob gtable_add_grob

---

makeci *Create confidence interval grob*

Description

Create confidence interval grob

Usage

```
makeci(
  est,
  lower,
  upper,
  pch,
  sizes = 1,
  gp = gpar(),
  t_height = NULL,
  xlim = c(0, 1),
  nudge_y = 0,
  name = NULL
)
```

Arguments

est	Point estimation. Can be a list for multiple columns and/or multiple groups. If the length of the list is larger than then length of ci_column, then the values reused for each column and considered as different groups.
lower	Lower bound of the confidence interval, same as est.
upper	Upper bound of the confidence interval, same as est.
pch	Numeric or character vector indicating what sort of plotting symbol to use. See pointsGrob.
sizes	Size of the point estimation box, can be a vector or a list. The value is a multiple of one line of text, so 1 draws a point as tall as the base_size of the theme. The same scale applies to the summary diamond. Values are used as they are, unless scale_sizes is used to read them as study weights; useful values are roughly between 0.2 and 1.5, and a warning is given when the plot is drawn if they are outside 0.1 to 2.
gp	Graphical parameters of gpar. Please refer to forest_theme for more details.
t_height	The height of the confidence interval line end vertices. If the value is NULL (default), no vertices will be drawn.
xlim	Limits for the x-axis as a vector of length 2, i.e. c(low, high). By default the minimum and maximum of the lower and upper values are used.

nudge_y	Vertical adjustment to nudge groups by, must be within 0 to 1. Defaults to 0; for grouped forest plots a value of 0 is bumped to 0.1 automatically so that group CIs do not overplot. Set explicitly to override.
name	Name of the grob.

Value

A gTree object

`make_boxplot` *Create horizontal boxplot grob*

Description

Create horizontal boxplot grob

Usage

```
make_boxplot(
  est,
  lower,
  upper,
  lowhinge,
  uphinge,
  hinge_height = 0.2,
  pch,
  sizes = 1,
  gp = gpar(),
  gp_box = gp,
  t_height = NULL,
  xlim = c(0, 1),
  nudge_y = 0
)
```

Arguments

est	Median value.
lower	Lower whisker.
upper	Upper whisker.
lowhinge	Lower hinge, a standard whisker will be drawn if this is missing.
uphinge	Upper hinge, a standard whisker will be drawn if this is missing.
hinge_height	Height of the hinge, default is 0.2.
pch	Numeric or character vector indicating what sort of plotting symbol to use. See pointsGrob.

sizes	Size of the point estimation box, can be a vector or a list. The value is a multiple of one line of text, so 1 draws a point as tall as the base_size of the theme. The same scale applies to the summary diamond. Values are used as they are, unless scale_sizes is used to read them as study weights; useful values are roughly between 0.2 and 1.5, and a warning is given when the plot is drawn if they are outside 0.1 to 2.
gp	Graphical parameters of gpar. Please refer to forest_theme for more details.
gp_box	Graphical parameters passed to the hinge, this will be passed to rectGrob. This does not support multiple groups.
t_height	Height of the whisker end vertices. If value is NULL (default), no vertices will be drawn.
xlim	Limits for the x axis as a vector of length 2, i.e. c(low, high).
nudge_y	Horizontal adjustment to nudge groups by, must be within 0 to 1.

Value

A gTree object

See Also

pointsGrob gpar rectGrob linesGrob segmentsGrob

Examples

```
library(grid)

# Function to calculate Box plot values
box_func <- function(x){
  iqr <- IQR(x)
  q3 <- quantile(x, probs = c(0.25, 0.5, 0.75), names = FALSE)
  c("min" = q3[1] - 1.5*iqr, "q1" = q3[1], "med" = q3[2],
    "q3" = q3[3], "max" = q3[3] + 1.5*iqr)
}
# Prepare data
val <- split(ToothGrowth$len, list(ToothGrowth$supp, ToothGrowth$dose))
val <- lapply(val, box_func)

dat <- do.call(rbind, val)
dat <- data.frame(Dose = row.names(dat),
                  dat, row.names = NULL)

dat$Box <- paste(rep(" ", 20), collapse = " ")

# Draw single group box plot
tm <- forest_theme(ci_Theight = 0.2)

p <- forest(dat[,c(1, 7)],
            est = dat$med,
            lower = dat$min,
            upper = dat$max,
```

```
            # sizes = sizes,
            fn_ci = make_boxplot,
            ci_column = 2,
            lowhinge = dat$q1,
            uphinge = dat$q3,
            hinge_height = 0.2,
            index_args = c("lowhinge", "uphinge"),
            gp_box = gpar(fill = "black", alpha = 0.4),
            style = tm
)
p

# Multiple group
# Prepare data
dat_oj <- dat[c(1, 3, 5),]
dat_vc <- dat[c(2, 4, 6), ]

dat <- data.frame(Dose = c(0.5, 1, 2))
dat$Box <- paste(rep(" ", 20), collapse = " ")

# Draw plot
tm <- forest_theme(ci_Theight = 0.2,
                   ci_pch = 3)

p <- forest(dat,
            est = list(dat_oj$med, dat_vc$med),
            lower = list(dat_oj$min, dat_vc$min),
            upper = list(dat_oj$max, dat_vc$max),
            fn_ci = make_boxplot,
            ci_column = 2,
            lowhinge = list(dat_oj$q1, dat_vc$q1),
            uphinge = list(dat_oj$q3, dat_vc$q3),
            hinge_height = 0.2,
            index_args = c("lowhinge", "uphinge"),
            style = tm
)

p
```

---

make_summary *Create pooled summary diamond shape*

Description

Create pooled summary diamond shape

Usage

```
make_summary(est, lower, upper, sizes = 1, gp, xlim, nudge_y = 0)
```

Arguments

est	Point estimation. Can be a list for multiple columns and/or multiple groups. If the length of the list is larger than then length of ci_column, then the values reused for each column and considered as different groups.
lower	Lower bound of the confidence interval, same as est.
upper	Upper bound of the confidence interval, same as est.
sizes	Size of the point estimation box, can be a vector or a list. The value is a multiple of one line of text, so 1 draws a point as tall as the base_size of the theme. The same scale applies to the summary diamond. Values are used as they are, unless scale_sizes is used to read them as study weights; useful values are roughly between 0.2 and 1.5, and a warning is given when the plot is drawn if they are outside 0.1 to 2.
gp	Graphical parameters of gpar. Please refer to forest_theme for more details.
xlim	Limits for the x-axis as a vector of length 2, i.e. c(low, high). By default the minimum and maximum of the lower and upper values are used.
nudge_y	Vertical adjustment to nudge groups by, must be within 0 to 1. Defaults to 0; for grouped forest plots a value of 0 is bumped to 0.1 automatically so that group CIs do not overplot. Set explicitly to override.

Value

A gTree object

---

print.forestplot *Draw plot*

Description

Print or draw forestplot.

Usage

```
## S3 method for class 'forestplot'
print(x, autofit = FALSE, ...)

## S3 method for class 'forestplot'
plot(x, autofit = FALSE, ...)
```

Arguments

x	forestplot to display
autofit	If true, the page is shared equally between the columns and between the rows of the plot. This will be deprecated, use fit of forest_style instead, which also works with ggplot2::ggsave and patchwork.
...	other arguments not used by this method

Value

Invisibly returns the original forestplot.

---

print.forest_style *Print a forest plot style*

Description

Show the settings of a style that are not the default ones.

Usage

```
## S3 method for class 'forest_style'
print(x, ...)
```

Arguments

x	A style created with forest_style.
...	other arguments not used by this method

Value

Invisibly returns the style.

---

scale_sizes *Scale point sizes by weights*

Description

Read the sizes given to forest as study weights and turn them into point sizes. The square root of the weights is taken first, so that the *area* of each point is proportional to its weight, before mapping onto range.

Usage

```
scale_sizes(plot, method = c("range", "proportional"), range = c(0.2, 0.8))
```

Arguments

plot	A forest plot object, see forest.
method	"range" (default) puts the smallest weight on range[1] and the largest on range[2], as metafor::forest.rma does with its plim. "proportional" keeps the areas proportional to the weights and only clamps the smallest points up, as meta::forest.meta does.
range	Numeric vector of length 2 giving the smallest and largest point size, as a multiple of one line of text.

Details

Weights are scaled jointly across all groups and CI columns so that the areas stay comparable between them; scale by hand if per-column control is wanted. Rows flagged by `is_summary` are held out of the scaling and drawn at `range[2]`, since a pooled total is not comparable with a study weight. This follows `meta`, whose pooled rows carry no study weight and end up the size of the largest study square, and `metafor`, which sizes its summary polygon from `efac` rather than from the weights.

Each call sets both the method and the range, and `method = NULL` turns the scaling off, so that `sizes` are used as they are.

Value

A forest plot object.

See Also

forest

Examples

```
library(grid)
# Read provided sample example data
dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))
dt <- dt[1:6, ]

# Add a blank column for the forest plot to display CI
dt$` ` <- paste(rep(" ", 20), collapse = " ")

# The weight of each study, here the inverse of the width of the CI
weights <- 1/(dt$hi - dt$low)

p <- forest(dt[, c("Subgroup", " ")],
            est = dt$est,
            lower = dt$low,
            upper = dt$hi,
            sizes = weights,        # weights, not sizes
            ci_column = 2,
            ref_line = 1)

# The area of each point is proportional to its weight
plot(scale_sizes(p, method = "range", range = c(0.2, 0.8)))

# `NULL` turns the scaling off, the values of `sizes` are then used as they are
plot(scale_sizes(p, method = NULL))
```

set_labs *Set the labels of a forest plot*

Description

Set the title, x-axis labels, footnote, arrow labels and legend text of a forest plot. Their look, including the justification of the title, the arrow type and the legend position, is set with forest_style.

Usage

```
set_labs(plot, title, xlab, footnote, arrow, legend_title, legend_labels)
```

Arguments

plot	A forest plot object, see forest.
title	The text for the title.
xlab	X-axis labels, put under the x-axis. A vector with one label for each CI column gives different labels to the columns, NA leaves a column without a label.
footnote	Footnote for the forest plot, aligned at the left bottom of the plot. Please adjust the line length with line breaks to avoid overlap with the arrows and/or x-axis.
arrow	Labels for the arrows under the x-axis, a vector of length two (left and right). A list with one pair of labels for each CI column gives different arrows to the columns, NA leaves a column without arrows.
legend_title	Title of the legend of a grouped forest plot, the default is "Group".
legend_labels	Legend labels, one for each group. Defaults to "Group 1", "Group 2", ...

Details

Arguments left out keep their current value, so the labels can be set in several calls. NULL removes a label, or for the legend goes back to the default text.

This builds the plot again, so it must be used before the plot is edited with edit_plot, add_text, insert_text, add_border or add_grob.

Value

A forest plot object.

See Also

forest forest_style

Examples

```
library(grid)
# Read provided sample example data
dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))
dt <- dt[1:6, ]

# Add a blank column for the forest plot to display CI
dt$` ` <- paste(rep(" ", 20), collapse = " ")

p <- forest(dt[, c("Subgroup", " ")],
            est = dt$est,
            lower = dt$low,
            upper = dt$hi,
            ci_column = 2,
            ref_line = 1)

p <- set_labs(p,
              title = "Subgroup analysis",
              xlab = "Hazard ratio",
              arrow = c("Placebo Better", "Treatment Better"),
              footnote = "This is the demo data.")
plot(p)

# Labels left out are kept, `NULL` removes a label
plot(set_labs(p, footnote = NULL))
```

`set_style` *Set the style of a forest plot*

Description

Change the look of a forest plot. The settings given in `...` update the current style of the plot: settings left out keep their current value, a `gpar` is merged into the current one and `NULL` goes back to the default. Like `set_labs`, this builds the plot again, so it must be used before the plot is edited with `edit_plot` and the other editing functions.

Usage

```
set_style(plot, style = NULL, ...)
```

Arguments

`plot`	A forest plot object, see `forest`.
`style`	A style created with `forest_style`, or a theme created with `forest_theme`, that replaces the current style of the plot. The settings in `...` are applied on top of it. `forest_style()` gives the default style.
`...`	Arguments of `forest_style` to change, for example `title = gpar(col = "red")`, `fit = "width"` or `core = list(...)`.

Value

A forest plot object.

See Also

forest_style forest

Examples

```
library(grid)
# Read provided sample example data
dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))
dt <- dt[1:6, ]

# Add a blank column for the forest plot to display CI
dt$` ` <- paste(rep(" ", 20), collapse = " ")

p <- forest(dt[, c("Subgroup", " ")],
            est = dt$est,
            lower = dt$low,
            upper = dt$hi,
            ci_column = 2,
            ref_line = 1,
            style = forest_style(base_size = 10, ref_line = gpar(col = "red")))

# Settings left out are kept, so the reference line stays red
p <- set_style(p, ci = gpar(col = "#4575b4"), title_just = "center")
plot(set_labs(p, title = "Subgroup analysis"))

# `NULL` goes back to the default, a style replaces the whole style
plot(set_style(p, ref_line = NULL))
plot(set_style(p, forest_style(base_size = 8)))

# Let the CI column take the width of the page it is drawn on
plot(set_style(p, fit = "width"))
```

---

set_xaxis *Set the x-axis*

Description

Set the limits, tick marks and scale of the x-axis of the CI columns, and add vertical lines to them.

Usage

```
set_xaxis(plot, xlim, ticks_at, ticks_digits, ticks_minor, x_trans, vline)
```

Arguments

plot	A forest plot object, see forest.
xlim	Limits for the x-axis as a vector of length 2, i.e. c(low, high). By default the minimum and maximum of the lower and upper values are used.
ticks_at	Tick mark positions. By default, ticks are computed automatically: pretty for linear axes and a decade-aware helper for log scales (e.g. 0.1, 1, 10, 100 for a wide log10 range).
ticks_digits	Number of digits for the tick labels. If an integer is given, for example 1L, trailing zeros after the decimal mark are dropped. Give a double, for example 1, to keep them. By default the number is calculated from the tick positions. Use a list to mix the two between CI columns, as a vector makes them all double.
ticks_minor	A numeric vector of positions to draw ticks without labels. It can be a superset of ticks_at or disjoint from it.
x_trans	Scale of the axis, one of "none" (default), "log", "log2" or "log10". Use "log" if the values are exponential, e.g. odds ratios or hazard ratios. The default reference line of forest is 1 for log scales and 0 otherwise.
vline	Numeric vector, positions of vertical lines drawn in addition to the reference line, on the original scale of the x-axis. Their look is set with vertline of forest_style. No lines are drawn by default.

Details

Arguments left out keep their current value, so the axis can be set in several calls, and NULL goes back to the default. A single value applies to all CI columns. To give the columns different settings, provide a list with one element for each CI column (a vector for ticks_digits and x_trans), where NA leaves a column at its default.

This builds the plot again, so it must be used before the plot is edited with edit_plot, add_text, insert_text, add_border or add_grob.

Value

A forest plot object.

See Also

forest set_labs forest_style

Examples

```
library(grid)
# Read provided sample example data
dt <- read.csv(system.file("extdata", "example_data.csv", package = "forestploter"))
dt <- dt[1:6, ]

# Add a blank column for the forest plot to display CI
dt$` ` <- paste(rep(" ", 20), collapse = " ")
```

```
p <- forest(dt[, c("Subgroup", " ")],
            est = dt$est,
            lower = dt$low,
            upper = dt$hi,
            ci_column = 2,
            ref_line = 1)

# Limits and tick marks, with a vertical line at 2
p <- set_xaxis(p, xlim = c(0, 4), ticks_at = c(0.5, 1, 2, 3), vline = 2)
plot(p)

# The axis can be set in several calls, `NULL` goes back to the default
p <- set_xaxis(p, ticks_digits = 1L)
p <- set_xaxis(p, vline = NULL)

# A log scale, where the reference line of `forest()` defaults to 1
plot(set_xaxis(p, x_trans = "log", xlim = c(0.25, 4), ticks_at = c(0.5, 1, 2, 4)))
```

Index

add_border, 2, *6*, *25*, *28*
add_grob, 3, *6*, *25*, *28*
add_text, 4, *6*, *25*, *28*
arrow, *10*, *15*

edit_plot, 5, *6*, *25, 26*, *28*
editGrob, *6*

forest, 6, *9–11*, *15*, *23–28*
forest_style, *6–8*, 9, *13*, *15, 16*, *22, 23*, *25–28*
forest_theme, *5–7*, 12, *18*, *20*, *22*, *26*

get_wh, *11*, 16
gpar, *3–6*, *9*, *11*, *13–15*, *17, 18*, *20*, *22*, *26*
gtable, *3–6*, *8*, *17*
gtable_add_grob, *3–5*, *18*

insert_text, *6*, 17, *25*, *28*

legendGrob, *10*, *14, 15*
linesGrob, *15*, *20*

make_boxplot, *7, 8*, 19
make_summary, *7, 8*, 21
makeci, *7, 8*, 18

plot.forestplot *(*print.forestplot*)*, 22
plotmath, *4*, *10*, *17*
pointsGrob, *15*, *18–20*
pretty, *28*
print.forest_style, 23
print.forestplot, 22

rectGrob, *5, 6*, *20*

scale_sizes, *6–8*, *18*, *20*, *22*, 23
segmentsGrob, *3*, *15*, *20*
set_labs, *6–9*, *11*, *15*, 25, *26*, *28*
set_style, *6–9*, *11*, *15*, 26
set_xaxis, *6–8*, *10*, 27

tableGrob, *8*, *11*, *15*
textGrob, *4–6*, *15*, *17, 18*

unit, *10*