Package ‘ggforestplotR’

March 25, 2026

Title Publication-Ready Forest Plots with 'ggplot2'

Version 0.1.0

Description Transform model coefficients into flexible forest plots using 'ggplot2'. Provides helpers to standardize coefficient data from a range of modelling workflows and render publication-ready forest plots with a consistent interface.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, patchwork, rlang

Suggests broom, dplyr, knitr, rmarkdown, survival, testthat (>= 3.0.0), tibble

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website pkgdown

URL `https://thatoneguy006.github.io/ggforestplotR/`, `https://github.com/thatoneguy006/ggforestplotR`

BugReports `https://github.com/thatoneguy006/ggforestplotR/issues`

NeedsCompilation no

Author Carson Richardson [aut, cre, cph]

Maintainer Carson Richardson <`carson.richardson@outlook.com`>

Repository CRAN

Date/Publication 2026-03-25 20:50:17 UTC

Contents

---

add_forest_table *Add a summary table to a forest plot*

Description

Compose a summary table onto a forest plot.

Usage

```
add_forest_table(
  plot = NULL,
  position = c("left", "right"),
  show_terms = TRUE,
  show_n = NULL,
  show_estimate = TRUE,
  show_p = FALSE,
  columns = NULL,
  term_header = "Term",
  n_header = "N",
  estimate_label = "Estimate",
  p_header = "P-value",
  digits = NULL,
  text_size = NULL,
  striped_rows = NULL,
  stripe_fill = NULL,
  stripe_colour = NULL,
  grid_lines = FALSE,
  grid_line_colour = "black",
  grid_line_size = 0.3,
  grid_line_linetype = 1
)
```

Arguments

`plot`	A plot created by `ggforestplot()`. Leave as `NULL` to use + `add_forest_table(...)` syntax.
`position`	Whether to place the table on the left or right of the forest plot.
`show_terms`	Whether to show the term column in the table.
`show_n`	Whether to show the `N` column. Defaults to `TRUE` when the underlying plot data includes an `n` column.
`show_estimate`	Whether to show the formatted estimate and confidence interval column.
`show_p`	Whether to display the p-value column.

`columns`	Optional explicit columns to display in the side table, in the order they should appear. Accepts names such as `"n"` and `"term"`, or positions `1:4` corresponding to `term`, `n`, `estimate`, and `p`. When supplied, this overrides the default `show_*` column selection.
`term_header`	Header text for the term column.
`n_header`	Header text for the `N` column.
`estimate_label`	Header label for the estimate column.
`p_header`	Header text for the p-value column.
`digits`	Number of digits used when formatting estimates and p-values. Defaults to `2`.
`text_size`	Text size for table contents. Defaults to `3.2`.
`striped_rows`	Whether to draw alternating row stripes behind the table. Defaults to the stripe setting used in `ggforestplot()`.
`stripe_fill`	Fill colour used for striped rows. Defaults to the stripe fill used in `ggforestplot()`.
`stripe_colour`	Outline colour for striped rows. Defaults to the stripe outline used in `ggforestplot()`.
`grid_lines`	Whether to draw black horizontal grid lines in the table.
`grid_line_colour`	Colour used for the table grid lines.
`grid_line_size`	Line width used for the table grid lines.
`grid_line_linetype`	Line type used for the table grid lines.

Value

A patchwork-composed plot containing the forest plot and side table, or a ggplot add-on object when `plot = NULL`.

Examples

```
coefs <- data.frame(
  term = c("Age", "BMI", "Treatment"),
  estimate = c(0.3, -0.2, 0.4),
  conf.low = c(0.1, -0.4, 0.2),
  conf.high = c(0.5, 0.0, 0.6),
  sample_size = c(120, 115, 98),
  p_value = c(0.012, 0.031, 0.004)
)

p <- ggforestplot(coefs, n = "sample_size", p.value = "p_value")
add_forest_table(
  p,
  position = "left",
  show_n = TRUE,
  show_p = TRUE,
  estimate_label = "Beta"
)

ggforestplot(coefs, n = "sample_size", p.value = "p_value") +
```

```
  add_forest_table(
    position = "right",
    show_n = TRUE,
    show_p = TRUE,
    estimate_label = "Beta"
  )
```

`add_split_table` *Add split tables around a forest plot*

Description

Compose split table blocks around a forest plot so that summary data appear on both sides of the plotting panel.

Usage

```
add_split_table(
  plot = NULL,
  show_terms = TRUE,
  show_n = NULL,
  show_estimate = TRUE,
  show_p = FALSE,
  left_columns = NULL,
  right_columns = NULL,
  term_header = "Term",
  n_header = "N",
  estimate_label = "Estimate",
  p_header = "P-value",
  digits = NULL,
  text_size = NULL,
  striped_rows = NULL,
  stripe_fill = NULL,
  stripe_colour = NULL,
  left_width = NULL,
  plot_width = NULL,
  right_width = NULL
)
```

Arguments

`plot`	A plot created by `ggforestplot()`. Leave as `NULL` to use `+ add_split_table(...)` syntax.
`show_terms`	Whether to include the term column in the default left-side selection when `left_columns` is not supplied.
`show_n`	Whether to include the `N` column in the default left-side selection when `left_columns` is not supplied. Defaults to `TRUE` when the underlying plot data include an `n` column.

`show_estimate`	Whether to include the formatted estimate and confidence interval column in the default right-side selection when `right_columns` is not supplied.
`show_p`	Whether to include the p-value column in the default right-side selection when `right_columns` is not supplied.
`left_columns`	Optional explicit columns to place on the left side of the forest plot. Accepts names such as `"term"` and `"n"`, or positions `1:4` corresponding to `term`, `n`, `estimate`, and `p`.
`right_columns`	Optional explicit columns to place on the right side of the forest plot. Accepts names such as `"estimate"` and `"p"`, or positions `1:4` corresponding to `term`, `n`, `estimate`, and `p`.
`term_header`	Header text for the term column.
`n_header`	Header text for the `N` column.
`estimate_label`	Header label for the estimate column.
`p_header`	Header text for the p-value column.
`digits`	Number of digits used when formatting estimates and p-values. Defaults to `2`.
`text_size`	Text size for table contents. Defaults to `3.2`.
`striped_rows`	Whether to draw alternating row stripes behind the split table layout. Defaults to the stripe setting used in `ggforestplot()`.
`stripe_fill`	Fill colour used for striped rows. Defaults to the stripe fill used in `ggforestplot()`.
`stripe_colour`	Outline colour for striped rows. Defaults to the stripe outline used in `ggforestplot()`.
`left_width`	Optional width allocated to the left table block. By default this is derived from the number of displayed left-side columns relative to `plot_width`.
`plot_width`	Optional width allocated to the forest plot panel. Defaults to `2.5`.
`right_width`	Optional width allocated to the right table block. By default this is derived from the number of displayed right-side columns relative to `plot_width`.

Value

A patchwork-composed plot containing a left table, the forest plot, and a right table, or a ggplot add-on object when `plot = NULL`.

Examples

```
coefs <- data.frame(
  term = c("Age", "BMI", "Treatment"),
  estimate = c(0.3, -0.2, 0.4),
  conf.low = c(0.1, -0.4, 0.2),
  conf.high = c(0.5, 0.0, 0.6),
  sample_size = c(120, 115, 98),
  p_value = c(0.012, 0.031, 0.004)
)

p <- ggforestplot(coefs, n = "sample_size", p.value = "p_value")
add_split_table(
  p,
```

```
    left_columns = c("term", "n"),
    right_columns = c("estimate", "p"),
    estimate_label = "HR"
  )

ggforestplot(coefs, n = "sample_size", p.value = "p_value") +
  add_split_table(
    left_columns = c(1, 2),
    right_columns = c(3, 4),
    estimate_label = "HR"
  )
```

`as_forest_data` *Standardize coefficient data for forest plots*

Description

Standardizes a coefficient table into the internal forest-plot data structure used throughout `ggforestplotR`.

Usage

```
as_forest_data(
  data,
  term,
  estimate,
  conf.low,
  conf.high,
  label = term,
  group = NULL,
  grouping = NULL,
  separate_groups = NULL,
  n = NULL,
  p.value = NULL,
  exponentiate = FALSE,
  sort_terms = c("none", "descending", "ascending")
)
```

Arguments

`data`	A data frame containing coefficient estimates and intervals.
`term`	Column name holding the model term identifier.
`estimate`	Column name holding the point estimate.
`conf.low`	Column name holding the lower confidence bound.
`conf.high`	Column name holding the upper confidence bound.
`label`	Optional column name used for the displayed row label.
`group`	Optional column name used for color-grouping multiple estimates per row.

`grouping`	Optional column name used to split rows into grouped plot sections.
`separate_groups`	Optional column name used to identify labeled variable blocks that can be outlined with separator lines.
`n`	Optional column name holding sample sizes or other N labels for table helpers.
`p.value`	Optional column name holding p-values.
`exponentiate`	Logical; if `TRUE`, require positive values for estimates and intervals.
`sort_terms`	How to sort rows: `"none"`, `"descending"`, or `"ascending"`.

Value

A standardized data frame ready for `ggforestplot()` and the table composition helpers.

Examples

```
raw <- data.frame(
  variable = c("Age", "BMI", "Treatment"),
  beta = c(0.10, -0.08, 0.34),
  lower = c(0.02, -0.16, 0.12),
  upper = c(0.18, 0.00, 0.56)
)

as_forest_data(
  data = raw,
  term = "variable",
  estimate = "beta",
  conf.low = "lower",
  conf.high = "upper"
)
```

`ggforestplot` *Draw a ggplot2 forest plot*

Description

Builds a forest plot from standardized coefficient data or directly from a fitted model.

Usage

```
ggforestplot(
  data,
  term = "term",
  estimate = "estimate",
  conf.low = "conf.low",
  conf.high = "conf.high",
  label = term,
  group = NULL,
```

```
  grouping = NULL,
  grouping_strip_position = c("left", "right"),
  separate_groups = NULL,
  n = NULL,
  p.value = NULL,
  exponentiate = FALSE,
  sort_terms = c("none", "descending", "ascending"),
  point_size = 2.3,
  point_shape = 19,
  line_size = 0.5,
  staple_width = 0.2,
  dodge_width = 0.6,
  separate_lines = FALSE,
  separator_line_linetype = 2,
  separator_line_colour = "black",
  separator_line_size = 0.4,
  striped_rows = FALSE,
  stripe_fill = "grey95",
  stripe_colour = NA,
  zero_line = TRUE,
  zero_line_linetype = 2,
  zero_line_colour = "grey60"
)
```

Arguments

`data`	Either a tidy coefficient data frame or a model object supported by `broom::tidy()`.
`term`	Column name holding the model term identifiers.
`estimate`	Column name holding the point estimates.
`conf.low`	Column name holding the lower confidence bounds.
`conf.high`	Column name holding the upper confidence bounds.
`label`	Optional column name used for the displayed row labels.
`group`	Optional column name used for color-grouping estimates.
`grouping`	Optional column name used to split rows into grouped plot sections.
`grouping_strip_position`	Positioning for grouped section strips.
`separate_groups`	Optional column name used to identify labeled variable blocks that can be outlined with grid lines.
`n`	Optional column name holding sample sizes or other N labels for table helpers.
`p.value`	Optional column name holding p-values.
`exponentiate`	Logical; if `TRUE`, transform the estimates and draw the axis on the log scale with the null line at 1.
`sort_terms`	How to sort rows: `"none"`, `"descending"`, or `"ascending"`.
`point_size`	Point size for coefficient markers.

`point_shape`	Shape used for coefficient markers.
`line_size`	Line width for confidence intervals.
`staple_width`	Width of the terminal staples on confidence interval lines.
`dodge_width`	Horizontal dodging used for grouped estimates.
`separate_lines`	Logical; if `TRUE`, draw grid lines around each labeled block identified by `separate_groups`.
`separator_line_linetype`	Line type used for separator lines.
`separator_line_colour`	Colour used for separator lines.
`separator_line_size`	Line width used for separator lines.
`striped_rows`	Logical; if `TRUE`, shade alternating rows.
`stripe_fill`	Fill color used for shaded rows.
`stripe_colour`	Border color for shaded rows.
`zero_line`	Logical; if `TRUE`, draw a null reference line.
`zero_line_linetype`	Line type for the null reference line.
`zero_line_colour`	Color for the null reference line.

Value

A `ggplot` object. Use standard `ggplot2` functions such as `ggplot2::labs()` for plot labels, and add composition helpers after styling the main plot.

Examples

```
coefs <- data.frame(
  term = c("Age", "BMI", "Treatment"),
  estimate = c(0.10, -0.08, 0.34),
  conf.low = c(0.02, -0.16, 0.12),
  conf.high = c(0.18, 0.00, 0.56)
)

ggforestplot(coefs)

ggforestplot(coefs, striped_rows = TRUE, point_shape = 17)
```

`tidy_forest_model` *Tidy a model object for forest plotting*

Description

Uses `broom::tidy()` to convert a fitted model into forest-plot data.

Usage

```
tidy_forest_model(
  model,
  conf.int = TRUE,
  conf.level = 0.95,
  exponentiate = FALSE,
  intercept = FALSE,
  sort_terms = c("none", "descending", "ascending")
)
```

Arguments

`model`	A fitted model object supported by `broom::tidy()`.
`conf.int`	Logical; if `TRUE`, request confidence intervals from `broom::tidy()`.
`conf.level`	Confidence level for intervals.
`exponentiate`	Logical; passed through to `broom::tidy()`.
`intercept`	Logical; if `FALSE`, drop the intercept term.
`sort_terms`	How to sort rows: `"none"`, `"descending"`, or `"ascending"`.

Value

A standardized coefficient data frame ready for `ggforestplot()`.

Examples

```
if (requireNamespace("broom", quietly = TRUE)) {
  fit <- lm(mpg ~ wt + hp + qsec, data = mtcars)
  tidy_forest_model(fit)

  set.seed(123)
  logit_data <- data.frame(
    age = rnorm(250, mean = 62, sd = 8),
    bmi = rnorm(250, mean = 28, sd = 4),
    treatment = factor(rbinom(250, 1, 0.45), labels = c("Control", "Treatment"))
  )
  linpred <- -9 + 0.09 * logit_data$age + 0.11 * logit_data$bmi +
    0.9 * (logit_data$treatment == "Treatment")
  logit_data$event <- rbinom(250, 1, plogis(linpred))
  logit_fit <- glm(event ~ age + bmi + treatment, data = logit_data, family = binomial())
```

```
  tidy_forest_model(logit_fit, exponentiate = TRUE)
}
```

Index