Package ‘forrest’

March 31, 2026

Title Publication-Ready Forest Plots

Version 0.3.0

Description Creates publication-ready forest plots from any tabular data containing point estimates and confidence intervals. Suitable for visualising results from regression models, meta-analyses, subgroup analyses, or any comparative study. Supports group and subgroup headings, summary estimates displayed as diamonds, grouped estimates with automatic colour and shape mapping, vertical dodging of multiple estimates within the same row, customisable text columns alongside the plot, and optional row striping. Provides a helper to export plots to PDF, PNG, SVG, or TIFF. Built on 'tinyplot' for clean, consistent visual styling with a minimal dependency footprint.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports tinyplot (>= 0.2.0)

Suggests broom, covr, data.table, knitr, marginaleffects, MatchIt, quarto, rmarkdown, tibble, testthat (>= 3.0.0)

Config/testthat/edition 3

Config/Needs/coverage covr

VignetteBuilder quarto, knitr

URL https://lorenzofabbri.github.io/forrest/, https://github.com/lorenzoFabbri/forrest

BugReports https://github.com/lorenzoFabbri/forrest/issues

NeedsCompilation no

Author Lorenzo Fabbri [aut, cre, cph]

Maintainer Lorenzo Fabbri <lorenzo.fabbri92sm@gmail.com>

Repository CRAN

Date/Publication 2026-03-31 15:10:02 UTC

Contents

`forrest` *Create a forest plot*

Description

Draws a publication-ready forest plot from a data frame. Each row represents one estimate — a study, a predictor, a model, a subgroup, or any other unit of analysis.

Usage

```
forrest(
  data,
  estimate,
  lower,
  upper,
  label = NULL,
  group = NULL,
  is_summary = NULL,
  weight = NULL,
  section = NULL,
  subsection = NULL,
  section_indent = TRUE,
  section_spacer = TRUE,
  section_cols = NULL,
  subsection_cols = NULL,
  ref_label = FALSE,
  ref_line = 0,
  log_scale = FALSE,
  xlim = NULL,
  xlab = "Estimate (95% CI)",
  title = NULL,
  header = NULL,
  cols = NULL,
  widths = NULL,
  stripe = FALSE,
  dodge = FALSE,
  pch = 15,
  shape = NULL,
  lwd = 2,
  cex = 1,
  col = NULL,
  cols_by_group = FALSE,
  legend_pos = "topright",
  legend_shape_pos = "bottomright",
  theme = "default",
  ...
)
```

Arguments

`data` A data frame, tibble, or data.table.

`estimate` Column name (string) for point estimates.

`lower` Column name (string) for lower confidence interval bounds.

`upper` Column name (string) for upper confidence interval bounds.

`label` Column name (string) for row labels displayed on the y-axis or in the left text panel. If `NULL`, row numbers are used.

`group` Column name (string) for a grouping variable. Rows that share a group value receive the same colour, and a legend is drawn automatically.

`is_summary` Column name (string) of a logical vector. Rows where this is `TRUE` are drawn as filled diamonds (e.g. pooled or overall estimates) rather than squares with whiskers.

`weight` Column name (string) of numeric row weights. When provided, point size scales as `cex * sqrt(weight / max(weight))`, so rows with larger weights appear with a bigger marker. Weights for summary rows are ignored (diamond size is fixed by `cex`).

`section` Column name (string) for a grouping variable that determines section structure. Whenever the value of this column changes (run-length boundary), a bold section header row is automatically inserted before the group. Row order is preserved; no automatic sorting is applied. See also `section_indent`, `section_spacer`, and `section_cols`.

`subsection` Column name (string) for a second-level grouping variable. Requires `section`. Inserts indented sub-headers beneath each section header. See also `subsection_cols`.

`section_indent` Logical. If `TRUE` (default), label values of data rows within a section are automatically indented by two spaces (four spaces for rows within a subsection).

`section_spacer` Logical. If `TRUE` (default), a blank spacer row is appended after the last row of each section.

`section_cols` Named character vector. Names must be a subset of the names of `cols`. Values are column names in `data` whose first non-NA entry in each section is shown in that section's header row. Columns not listed here display `""` in the header row. Use this to show section-level summaries (e.g. `"k = 3 studies"`) next to the section header.

`subsection_cols`
Like `section_cols` but for subsection header rows.

`ref_label` Logical. When `TRUE` and `section` is provided, rows with `NA` estimates that are present in the original data (i.e. reference category rows, not auto-generated headers) have `" (Ref.)"` appended to their label. Default is `FALSE`.

`ref_line` Numeric. Position of the vertical reference line (e.g. `0` for differences, `1` for ratio measures on the natural scale). Set to `NULL` to suppress. Default is `0`.

`log_scale` Logical. If `TRUE`, apply a log transformation to the x-axis. Useful when plotting odds ratios, hazard ratios, or risk ratios on the natural scale. Default is `FALSE`.

`xlim` Numeric vector of length 2 giving x-axis limits. Computed from the data when `NULL` (default). Confidence intervals that extend beyond `xlim` are clipped at the axis boundary and an arrow is drawn to indicate truncation.

`xlab` Label for the x-axis. Default is `"Estimate (95% CI)"`.

`title` Plot title. Default is `NULL` (no title).

`header` Optional header string placed above the label column. When `cols` is provided this appears above the left text panel; otherwise it is drawn above the topmost row on the y-axis.

`cols` Named character vector specifying extra text columns to display to the right of the plot. Names become column headers; values are column names in `data`. Example: `cols = c("OR (95\\% CI)" = "or_ci")`.

`widths` Numeric vector of relative panel widths. When `cols` is `NULL`, ignored. Otherwise, length must equal `2 + length(cols)`: label panel, plot panel, then one entry per extra column. Sensible defaults are chosen automatically.

`stripe` Logical. If `TRUE`, alternate rows are shaded with a light grey background to improve readability. Default is `FALSE`.

`dodge` Logical or positive numeric. When `TRUE` (or a positive number), consecutive rows that share the same `label` value are grouped together and their confidence intervals are drawn with a small vertical offset so that they do not overlap. The shared label is displayed once at the centre of the group. Use together with `group` (for colour) and/or `shape` (for point characters) to distinguish the overlaid series. A numeric value sets the offset between rows in a group directly (in y-axis units); `TRUE` uses a default of `0.25`. Default is `FALSE`.

`pch` Point character for non-summary rows. Default is `15` (filled square). When `shape` is provided, `pch` is used only as a fallback for rows whose shape value is `NA`.

`shape` Column name (string) for a shape variable. When provided, different values of the column are rendered with different point characters and a shape legend is drawn. Use together with `group` to distinguish two categorical dimensions simultaneously (e.g. colour = time period, shape = sex).

`lwd` Line width for confidence interval whiskers. Default is `2`.

`cex` Point size multiplier. Default is `1`.

`col` Colour or character vector of colours. When `NULL` (default) and `group` is specified, the Okabe-Ito colorblind-safe palette is used. When `NULL` and no `group`, a single dark colour is used.

`cols_by_group` Logical. Relevant only when dodge is active. When `TRUE`, each text column in `cols` is collapsed to one value per label group: the first non-empty entry within the group is displayed at the group centre y position. This produces a wide-format text table with one row per label and one column per condition. Populate each text column so that the value is non-empty only for the matching condition row and empty ("") for all others; `forrest()` picks up the right value automatically. When `FALSE` (default), text values are drawn at each individual row's dodged y position, keeping them aligned with their CI whiskers.

`legend_pos` Position of the colour legend when `group` is supplied. Passed to `legend()`. Use `NULL` to suppress. Default is `"topright"`.

`legend_shape_pos`
Position of the shape legend when `shape` is supplied. Passed to `legend()`. Use `NULL` to suppress. Default is `"bottomright"`.

`theme` Visual theme name (`"default"`, `"minimal"`, `"classic"`) or a named list of style overrides. Default is `"default"`.

`...` Graphical parameters forwarded to the internal tinyplot call (e.g. `cex.axis`, `cex.lab`).

Details

Rows with `NA` estimates are treated as reference-category rows: they produce no point or confidence interval, and their label is rendered in regular (non-bold) font. To create section headers and spacers automatically, use the `section` (and optionally `subsection`) arguments instead of inserting NA rows by hand.

Value

Invisibly returns `NULL`. Called for its side effect of producing a plot.

Examples

```
# Basic forest plot: linear model coefficients
dat <- data.frame(
  predictor = c("Age (per 10 y)", "Female sex",
                "BMI (per 5 kg/m\u00b2)", "Current smoker"),
  estimate  = c(0.42, -0.38,  0.19, -0.31),
  lower     = c(0.22, -0.56, -0.02, -0.51),
  upper     = c(0.62, -0.20,  0.40, -0.11)
)
forrest(dat,
  estimate = "estimate",
  lower    = "lower",
  upper    = "upper",
  label    = "predictor",
  xlab     = "Regression coefficient (95% CI)"
)

# Section headers from a grouping column
dat2 <- data.frame(
  domain    = c("Lifestyle", "Lifestyle", "Clinical", "Clinical"),
  predictor = c("Physical activity", "Diet quality",
                "BMI (per 5 kg/m\u00b2)", "Systolic BP (per 10 mmHg)"),
  estimate  = c(-0.31, -0.18,  0.19,  0.25),
  lower     = c(-0.51, -0.36, -0.02,  0.08),
  upper     = c(-0.11, -0.00,  0.40,  0.42)
)
forrest(dat2,
  estimate  = "estimate",
  lower     = "lower",
  upper     = "upper",
  label     = "predictor",
  section   = "domain",
  xlab      = "Regression coefficient (95% CI)"
)
```

save_forrest *Save a forest plot to a file*

Description

Writes the plot produced by forrest() (or any base-R plotting code) to a file. The graphics device is inferred from the file extension.

Usage

```
save_forrest(file, plot, width = 7, height = 5, dpi = 300, bg = "white")
```

Arguments

file	Output file path. Supported extensions: .pdf, .png, .svg, .tiff.
plot	A zero-argument function whose body calls forrest(). Evaluated inside the open graphics device.
width	Plot width in inches. Default 7.
height	Plot height in inches. Default 5.
dpi	Resolution in dots per inch for raster formats (.png, .tiff). Ignored for vector formats (.pdf, .svg). Default 300.
bg	Background colour. Default "white".

Value

Invisibly returns file.

Examples

```
dat <- data.frame(
  label    = c("Age (per 10 y)", "Female sex", "Current smoker"),
  estimate = c(0.42, -0.38, -0.31),
  lower    = c(0.22, -0.56, -0.51),
  upper    = c(0.62, -0.20, -0.11)
)
tmp <- tempfile(fileext = ".pdf")
```

```
save_forrest(tmp, function() {
  forrest(
    dat,
    estimate = "estimate",
    lower    = "lower",
    upper    = "upper",
    label    = "label",
    xlab     = "Regression coefficient (95% CI)"
  )
})
```

Index