

Package: ggadjustedforest (via r-universe)

July 17, 2026

Title Publication Ready Forest Plots for Estimand of Interest

Version 0.1.0

Description Creates publication-quality forest plots and effect-size tables that display only the unadjusted and adjusted estimates for a user-specified exposure variable of interest, hiding confounder coefficients in accordance with causal inference principles. Supports logistic, linear, Poisson, and Cox proportional hazards models, with optional cumulative-adjustment visualisation. Built on 'ggplot2' and follows the tidyverse design philosophy.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

RoxygenNote 8.0.0

Depends R (>= 4.1.0)

Imports ggplot2 (>= 3.4.0), dplyr (>= 1.1.0), rlang (>= 1.1.0), broom (>= 1.0.0), survival (>= 3.5.0), patchwork (>= 1.2.0), scales (>= 1.3.0), tibble (>= 3.2.0)

Suggests sandwich, lmttest, testthat (>= 3.0.0), knitr, rmarkdown, finalfit

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/kriz98/gg_adjusted_forest

BugReports https://github.com/kriz98/gg_adjusted_forest/issues

NeedsCompilation no

Author Chris Varghese [aut, cre]

Maintainer Chris Varghese <chris.varghese@auckland.ac.nz>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-17 12:40:02 UTC

RemoteUrl <https://github.com/cran/ggadjustedforest>

RemoteRef HEAD
RemoteSha 0e71abe1513ddd31aebf61430ebb8710b65f826f

Contents

forest_table	2
gg_adjusted_forest	4
plot.ggadjustedforest	6
print.ggadjustedforest	7
Index	8

forest_table	<i>Extract a formatted effect-size table without generating a plot</i>
--------------	--

Description

A convenience wrapper around gg_adjusted_forest() that returns only the formatted table. Useful when you want numeric summaries without producing a graphic.

Usage

```
forest_table(  
  data,  
  outcome,  
  exposure,  
  covariates = NULL,  
  model_type = "logistic",  
  cumulative = FALSE,  
  cumulative_labels = NULL,  
  conf_level = 0.95,  
  time_var = NULL,  
  event_var = NULL,  
  strata = NULL,  
  cluster = NULL,  
  weights = NULL,  
  table_digits = 2  
)
```

Arguments

data	A data frame containing all variables.
outcome	Character string. Name of the outcome variable (ignored for Cox models - use time_var and event_var instead).
exposure	Character string. Name of the exposure variable of interest.

covariates	Character vector of confounder/covariate names. In non-cumulative mode all covariates are added together; in cumulative mode they are added one at a time in the order supplied. Default NULL produces only the unadjusted estimate.
model_type	Character. One of "logistic" (default), "linear", "poisson", or "coxph".
cumulative	Logical. If TRUE, fit models that progressively add one covariate at a time and show each step as a separate row. Default FALSE.
cumulative_labels	Optional named character vector to rename the cumulative model labels. Names should match the auto-generated labels (e.g., "+ age", "+ age + sex"); values are the replacement labels.
conf_level	Numeric. Confidence level for intervals. Default 0.95.
time_var	Character. Name of the time variable (Cox model only).
event_var	Character. Name of the event indicator variable (Cox model only; should be 0/1 or logical).
strata	Character. Name of a stratification variable for Cox models. Default NULL.
cluster	Character. Name of a clustering variable for cluster-robust standard errors. Requires the sandwich and lme4 packages. Default NULL.
weights	Character. Name of a survey/frequency weight variable. Default NULL.
table_digits	Integer. Number of decimal places in the table. Default 2.

Value

A data frame with columns:

model Row label (e.g., "Unadjusted", "Adjusted").

estimate Point estimate (formatted character).

ci Confidence interval as a character string (e.g., "0.95–1.42").

formatted Combined estimate and CI (e.g., "1.15 (0.95–1.42)").

p.value Formatted p-value character string.

n Number of observations.

Examples

```
data(mtcars)
mtcars$am <- as.integer(mtcars$am)
forest_table(
  data      = mtcars,
  outcome   = "am",
  exposure  = "hp",
  covariates = c("wt", "cyl"),
  model_type = "logistic"
)
```

gg_adjusted_forest	<i>Forest plot of unadjusted and adjusted effects for a single exposure</i>
--------------------	---

Description

Creates a publication-quality forest plot showing only the unadjusted and adjusted (or cumulatively adjusted) effect estimates for a specified exposure variable, hiding confounder coefficients in accordance with causal inference principles.

Usage

```
gg_adjusted_forest(
  data,
  outcome,
  exposure,
  covariates = NULL,
  model_type = "logistic",
  cumulative = FALSE,
  cumulative_labels = NULL,
  effect_label = NULL,
  title = NULL,
  ref_line = NULL,
  point_size = 4,
  point_shape = 15,
  line_size = 0.7,
  color = "black",
  colour = NULL,
  vline_color = "grey50",
  vline_linetype = "dashed",
  x_limits = NULL,
  x_breaks = NULL,
  log_scale = TRUE,
  conf_level = 0.95,
  time_var = NULL,
  event_var = NULL,
  strata = NULL,
  cluster = NULL,
  weights = NULL,
  show_table = TRUE,
  table_digits = 2
)
```

Arguments

data	A data frame containing all variables.
outcome	Character string. Name of the outcome variable (ignored for Cox models - use time_var and event_var instead).

exposure	Character string. Name of the exposure variable of interest.
covariates	Character vector of confounder/covariate names. In non-cumulative mode all covariates are added together; in cumulative mode they are added one at a time in the order supplied. Default NULL produces only the unadjusted estimate.
model_type	Character. One of "logistic" (default), "linear", "poisson", or "coxph".
cumulative	Logical. If TRUE, fit models that progressively add one covariate at a time and show each step as a separate row. Default FALSE.
cumulative_labels	Optional named character vector to rename the cumulative model labels. Names should match the auto-generated labels (e.g., "+ age", "+ age + sex"); values are the replacement labels.
effect_label	Character. X-axis label. Defaults to "Odds Ratio (95 \% CI)" for logistic, "Risk Ratio (95 \% CI)" for Poisson, "Hazard Ratio (95 \% CI)" for Cox, and "Coefficient (95 \% CI)" for linear.
title	Character. Plot title. Default NULL (no title).
ref_line	Numeric. Position of the vertical reference line. Defaults to 1 for ratio models and 0 for linear.
point_size	Numeric. Size of the point estimate symbol. Default 4.
point_shape	Integer. ggplot2 shape code. Default 15 (filled square).
line_size	Numeric. Thickness of the CI lines. Default 0.7.
color	Character. Colour for points and CI lines. Default "black".
colour	Alias for color (British English spelling).
vline_color	Character. Colour of the reference line. Default "grey50".
vline_linetype	Character. Linetype of the reference line. Default "dashed".
x_limits	Numeric vector of length 2. Manual x-axis limits. Default NULL (automatic).
x_breaks	Numeric vector. Manual x-axis break positions. Default NULL (automatic).
log_scale	Logical. Use log scale on the x-axis for ratio models. Default TRUE.
conf_level	Numeric. Confidence level for intervals. Default 0.95.
time_var	Character. Name of the time variable (Cox model only).
event_var	Character. Name of the event indicator variable (Cox model only; should be 0/1 or logical).
strata	Character. Name of a stratification variable for Cox models. Default NULL.
cluster	Character. Name of a clustering variable for cluster-robust standard errors. Requires the sandwich and lmtest packages. Default NULL.
weights	Character. Name of a survey/frequency weight variable. Default NULL.
show_table	Logical. Combine the forest plot with a formatted table panel (using patchwork). Default TRUE.
table_digits	Integer. Number of decimal places in the table. Default 2.

Value

An object of class `ggadjustedforest` (a list) with components:

`plot` The combined **ggplot2/patchwork** plot object. When `show_table = FALSE` this is just the forest plot.

`table` A data frame with columns `model`, `estimate`, `conf.low`, `conf.high`, `p.value`, and `n`.

`formatted_table` A data frame with a formatted column containing strings like "1.23 (1.01-1.55)".

Examples

```
# Logistic regression example
data(mtcars)
mtcars$am <- as.integer(mtcars$am) # binary outcome
result <- gg_adjusted_forest(
  data      = mtcars,
  outcome   = "am",
  exposure  = "hp",
  covariates = c("wt", "cyl"),
  model_type = "logistic",
  title     = "Effect of Horsepower on Transmission Type"
)
result$table

# Cumulative adjustment
result2 <- gg_adjusted_forest(
  data      = mtcars,
  outcome   = "am",
  exposure  = "hp",
  covariates = c("wt", "cyl"),
  cumulative = TRUE
)
result2$table
```

`plot.ggadjustedforest` *Plot method for ggadjustedforest objects*

Description

Plot method for `ggadjustedforest` objects

Usage

```
## S3 method for class 'ggadjustedforest'
plot(x, ...)
```

Arguments

`x` An object of class `ggadjustedforest`.
`...` Additional arguments (currently ignored).

Value

Invisibly returns the plot object.

`print.ggadjustedforest`

Print method for ggadjustedforest objects

Description

Print method for ggadjustedforest objects

Usage

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

Arguments

<code>x</code>	An object of class <code>ggadjustedforest</code> .
<code>...</code>	Additional arguments (currently ignored).

Value

Invisibly returns `x`.

Index

`forest_table`, [2](#)

`gg_adjusted_forest`, [4](#)

`plot.ggadjustedforest`, [6](#)

`print.ggadjustedforest`, [7](#)