

Package ‘forestmodel’

July 19, 2020

Encoding UTF-8

Type Package

Title Forest Plots from Regression Models

Version 0.6.2

Date 2020-07-19

Author Nick Kennedy <r@nick-kennedy.com>

Maintainer Nick Kennedy <r@nick-kennedy.com>

Description Produces forest plots using 'ggplot2' from models produced by functions such as stats::lm(), stats::glm() and survival::coxph().

License GPL-2

LazyData TRUE

Depends R (>= 3.3.0), ggplot2 (>= 3.1.0)

Imports dplyr (>= 0.8.0), broom (>= 0.5.0), rlang (>= 0.3.0), tibble (>= 1.4.2)

Suggests survival, metafor, labelled

RoxygenNote 7.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2020-07-19 11:50:03 UTC

R topics documented:

default_forest_panels	2
forest_breaks	2
forest_model	3
forest_model_format_options	6
forest_panel	6
forest_panels	7
forest_rma	8
panel_forest_plot	10
theme_forest	11

Index**12**

 default_forest_panels *Default panels for forest_model*

Description

Default panels for forest_model

Usage

```
default_forest_panels(
  model = NULL,
  factor_separate_line = FALSE,
  measure = NULL,
  trans_char = "I"
)
```

Arguments

model	model object to guess label and determine defaults
factor_separate_line	changes defaults for widths of variable depending on whether factors have their name on separate line
measure	label for main forest plot
trans_char	character representation of transform for axes

Value

'list' ready to be passed to 'forest_model'

 forest_breaks *Calculate default breaks for limits*

Description

This function does not work as well as grDevices::axisTicks and so that should be used instead.

Usage

```
forest_breaks(limits, trans = I)
```

Arguments

limits	limits of plot
trans	transformation that will be used on the limits

Value

a vector with breaks ready to pass to [panel_forest_plot](#)

forest_model	<i>Produce a forest plot based on a regression model</i>
--------------	--

Description

Produce a forest plot based on a regression model

Usage

```
forest_model(
  model,
  panels = default_forest_panels(model, factor_separate_line = factor_separate_line),
  covariates = NULL,
  exponentiate = NULL,
  funcs = NULL,
  factor_separate_line = FALSE,
  format_options = forest_model_format_options(),
  theme = theme_forest(),
  limits = NULL,
  breaks = NULL,
  return_data = FALSE,
  recalculate_width = TRUE,
  recalculate_height = TRUE,
  model_list = NULL,
  merge_models = FALSE,
  exclude_infinite_cis = TRUE
)
```

Arguments

model	regression model produced by lm , glm , coxph
panels	list with details of the panels that make up the plot (See Details)
covariates	a character vector optionally listing the variables to include in the plot (defaults to all variables)
exponentiate	whether the numbers on the x scale should be exponentiated for plotting
funcs	optional list of functions required for formatting <code>panels\$display</code>
factor_separate_line	whether to show the factor variable name on a separate line
format_options	formatting options as a list as generated by forest_model_format_options
theme	theme to apply to the plot
limits	limits of the forest plot on the X-axis (taken as the range of the data by default)

breaks	breaks to appear on the X-axis (note these will be exponentiated if <code>exponentiate == TRUE</code>)
return_data	return the data to produce the plot as well as the plot itself
recalculate_width	TRUE to recalculate panel widths using the current device or the desired plot width in inches
recalculate_height	TRUE to shrink text size using the current device or the desired plot height in inches
model_list	list of models to incorporate into a single forest plot
merge_models	if 'TRUE', merge all models in one section.
exclude_infinite_cis	whether to exclude points and confidence intervals that go to positive or negative infinity from plotting. They will still be displayed as text. Defaults to TRUE, since otherwise plot is malformed

Details

This function takes the model output from one of the common model functions in R (e.g. `lm`, `glm`, `coxph`). If a label attribute was present on any of the columns in the original data (e.g. from the `labelled` package), this label is used in preference to the column name.

The `panels` parameter is a list of lists each of which have an element `width` and, optionally, `item`, `display`, `display_na`, `heading`, `hjust` and `fontface`. `item` can be "forest" for the forest plot (exactly one required) or "vline" for a vertical line. `display` indicates which column to display as text. It can be a quoted variable name or a formula. The column `display` can include the standard ones produced by `tidy` and in addition `variable` (the term in the model; for factors this is the bare variable without the level), `level` (the level of factors), `reference` (TRUE for the reference level of a factor). For `coxph` models, there will also be `n_events` for the number of events in the group with that level of the factor and `person_time` for the person-time in that group. The function `trans` is defined to be the transformation between the coefficients and the scales (e.g. `exp`). Other functions not in base R can be provided as a list with the parameter `funcs`. `display_na` allows for an alternative display for NA terms within `estimate`.

Value

A `ggplot` ready for display or saving, or (with `return_data == TRUE`, a list with the parameters to call `panel_forest_plot` in the element `plot_data` and the `ggplot` itself in the element `plot`)

Examples

```
library("survival")
library("dplyr")
pretty_lung <- lung %>%
  transmute(time,
            status,
            Age = age,
            Sex = factor(sex, labels = c("Male", "Female"))),
```

```

      ECOG = factor(lung$ph.ecog),
      `Meal Cal` = meal.cal
    )

print(forest_model(coxph(Surv(time, status) ~ ., pretty_lung)))

# Example with custom panels

panels <- list(
  list(width = 0.03),
  list(width = 0.1, display = ~variable, fontface = "bold", heading = "Variable"),
  list(width = 0.1, display = ~level),
  list(width = 0.05, display = ~n, hjust = 1, heading = "N"),
  list(width = 0.05, display = ~n_events, width = 0.05, hjust = 1, heading = "Events"),
  list(
    width = 0.05,
    display = ~ replace(sprintf("%0.1f", person_time / 365.25), is.na(person_time), ""),
    heading = "Person-\nYears", hjust = 1
  ),
  list(width = 0.03, item = "vline", hjust = 0.5),
  list(
    width = 0.55, item = "forest", hjust = 0.5, heading = "Hazard ratio", linetype = "dashed",
    line_x = 0
  ),
  list(width = 0.03, item = "vline", hjust = 0.5),
  list(width = 0.12, display = ~ ifelse(reference, "Reference", sprintf(
    "%0.2f (%0.2f, %0.2f)",
    trans(estimate), trans(conf.low), trans(conf.high)
  )), display_na = NA),
  list(
    width = 0.05,
    display = ~ ifelse(reference, "", format.pval(p.value, digits = 1, eps = 0.001)),
    display_na = NA, hjust = 1, heading = "p"
  ),
  list(width = 0.03)
)
forest_model(coxph(Surv(time, status) ~ ., pretty_lung), panels)

data_for_lm <- tibble(
  x = rnorm(100, 4),
  y = rnorm(100, 3, 0.5),
  z = rnorm(100, 2, 2),
  outcome = 3 * x - 2 * y + 4 * z + rnorm(100, 0, 0.1)
)

print(forest_model(lm(outcome ~ ., data_for_lm)))

data_for_logistic <- data_for_lm %>% mutate(
  outcome = (0.5 * (x - 4) * (y - 3) * (z - 2) + rnorm(100, 0, 0.05)) > 0.5
)

print(forest_model(glm(outcome ~ ., binomial(), data_for_logistic)))

```

forest_model_format_options
Create format options for forest_model

Description

Create format options for forest_model

Usage

```
forest_model_format_options(  
  colour = "black",  
  color = NULL,  
  shape = 15,  
  text_size = 5,  
  point_size = 5,  
  banded = TRUE  
)
```

Arguments

colour	colour of the point estimate and error bars
color	alias for colour
shape	shape of the point estimate
text_size	text size in mm
point_size	point size
banded	whether to show light grey bands behind alternate rows

Value

list of format options

forest_panel *Create definition of a panel for forest_model*

Description

Create definition of a panel for forest_model

Usage

```
forest_panel(
  width,
  item = c("", "forest", "vline"),
  display = NULL,
  display_na = NULL,
  hjust = NULL,
  heading = NULL,
  fontface = NULL,
  linetype = NULL,
  line_x = NULL,
  parse = NULL,
  width_group = NULL
)
```

Arguments

width	relative width of the panel
item	specification of which type of item to use; overridden if display is not missing
display	bare expression that specifies the variable or expression to display
display_na	what to display if a value is NA
hjust	horizontal justification
heading	heading to be used (defaults to the variable name)
fontface	fontface to use
linetype	line type to use
line_x	position for dashed line in forest plot
parse	whether text should be parsed as expressions
width_group	grouping used when recalculating widths of panels

Value

panel definition as a list

forest_panels	<i>Generate panels for forest plots</i>
---------------	---

Description

Generate panels for forest plots

Usage

```
forest_panels(..., margin = 0.03)
```

Arguments

... panels to variables in data
margin margin to leave at left and right edges

Value

a panels list ready for `forest_model` or `forest_rma`

forest_rma	<i>Generate a forest plot from a meta-analysis</i>
------------	--

Description

Generate a forest plot from a meta-analysis

Usage

```
forest_rma(
  model,
  panels = NULL,
  study_labels = NULL,
  additional_data = NULL,
  point_size = NULL,
  model_label = NULL,
  show_individual_studies = TRUE,
  show_model = TRUE,
  show_stats = list(`I^2` = rlang::quo(sprintf("%0.1f%%", I2)), p =
    rlang::quo(format.pval(QEp, digits = 4, eps = 1e-04, scientific = 1))),
  trans = I,
  funcs = NULL,
  format_options = forest_model_format_options(),
  theme = theme_forest(),
  limits = NULL,
  breaks = NULL,
  return_data = FALSE,
  recalculate_width = TRUE,
  recalculate_height = TRUE
)
```

Arguments

model a single `rma` object or a list of them
panels list with details of the panels that make up the plot (See Details)
study_labels a character vector of study labels or list of character vectors the same length as model

<code>additional_data</code>	a data.frame of additional data that can be referenced for the data shown in the panels of the forest plot
<code>point_size</code>	a numeric vector with the point sizes for the individual studies, or a single value used for all studies, or a list of numeric vectors if more than one model is to be plotted
<code>model_label</code>	a single model label or character vector of model labels the same length as <code>model</code>
<code>show_individual_studies</code>	whether to show the individual studies (the default) or just the summary diamond
<code>show_model</code>	a logical value, if 'TRUE', show model result, otherwise only show forest plots for studies
<code>show_stats</code>	a list of stats to show at the bottom of the forest plot for e.g. heterogeneity
<code>trans</code>	an optional transform function used on the numeric data for plotting the axes
<code>funcs</code>	optional list of functions required for formatting panels's display
<code>format_options</code>	formatting options as a list as generated by forest_model_format_options
<code>theme</code>	theme to apply to the plot
<code>limits</code>	limits of the forest plot on the X-axis (taken as the range of the data by default)
<code>breaks</code>	breaks to appear on the X-axis (note these will be exponentiated if <code>exponentiate == TRUE</code>)
<code>return_data</code>	return the data to produce the plot as well as the plot itself
<code>recalculate_width</code>	TRUE to recalculate panel widths using the current device or the desired plot width in inches
<code>recalculate_height</code>	TRUE to shrink text size using the current device or the desired plot height in inches

Details

This produces a forest plot using the [rma](#)

Value

plot

Examples

```
if (require("metafor")) {
  data("dat.bcg")
  dat <- escalc(measure = "RR", ai = tpos, bi = tneg, ci = cpos, di = cneg, data = dat.bcg)
  model <- rma(yi, vi, data = dat)

  print(forest_rma(model,
    study_labels = paste(dat.bcg$author, dat.bcg$year),
    trans = exp
  ))
}
```

```

print(forest_rma(model,
  panels = forest_panels(
    Study = ~study,
    N = ~n, ~vline, `Log Relative Risk` = ~ forest(line_x = 0),
    ~ spacer(space = 0.10),
    ~ sprintf("%0.3f (%0.3f, %0.3f)", estimate, conf.low, conf.high)
  ),
  study_labels = paste(dat.bcg$author, dat.bcg$year),
  trans = exp
))
}

```

panel_forest_plot *Plot a forest plot with panels of text*

Description

Plot a forest plot with panels of text

Usage

```

panel_forest_plot(
  forest_data,
  mapping = aes(estimate, xmin = conf.low, xmax = conf.high),
  panels = default_forest_panels(),
  trans = I,
  funcs = NULL,
  format_options = list(colour = "black", shape = 15, banded = TRUE, text_size = 5,
    point_size = 5),
  theme = theme_forest(),
  limits = NULL,
  breaks = NULL,
  recalculate_width = TRUE,
  recalculate_height = TRUE,
  exclude_infinite_cis = TRUE
)

```

Arguments

forest_data	data.frame with the data needed for both the plot and text
mapping	mapping aesthetic created using aes
panels	list with details of the panels that make up the plot (See Details)
trans	transform for scales
funcs	optional list of functions required for formatting panels\$display
format_options	formatting options as a list as generated by forest_model_format_options

theme	theme to apply to the plot
limits	limits of the forest plot on the X-axis (taken as the range of the data by default)
breaks	breaks to appear on the X-axis (note these will be exponentiated if <code>exponentiate == TRUE</code>)
recalculate_width	TRUE to recalculate panel widths using the current device or the desired plot width in inches
recalculate_height	TRUE to shrink text size using the current device or the desired plot height in inches
exclude_infinite_cis	whether to exclude points and confidence intervals that go to positive or negative infinity from plotting. They will still be displayed as text. Defaults to TRUE, since otherwise plot is malformed

Value

A ggplot ready for display or saving

theme_forest	<i>Default forest theme</i>
--------------	-----------------------------

Description

Default forest theme

Usage

```
theme_forest()
```

Value

a theme object for use with ggplot2

Index

`aes`, [10](#)

`coxph`, [3](#), [4](#)

`default_forest_panels`, [2](#)

`forest_breaks`, [2](#)

`forest_model`, [3](#), [8](#)

`forest_model_format_options`, [3](#), [6](#), [9](#), [10](#)

`forest_panel`, [6](#)

`forest_panels`, [7](#)

`forest_rma`, [8](#), [8](#)

`glm`, [3](#), [4](#)

`lm`, [3](#), [4](#)

`panel_forest_plot`, [3](#), [4](#), [10](#)

`rma`, [8](#), [9](#)

`theme_forest`, [11](#)

`tidy`, [4](#)