Package ‘forestmodel’

October 13, 2022

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
**default_forest_panels**  
Default panels for forest_model

**Description**

Default panels for forest_model

**Usage**

```r
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**

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

**Arguments**

- `limits`: limits of plot
- `trans`: transformation that will be used on the limits
forest_model

Value

a vector with breaks ready to pass to panel_forest_plot

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

Value

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

Examples

```r
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-Years", 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**

```r
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

```r
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**

Generate panels for forest plots

Description

Generate panels for forest plots

Usage

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

Generate a forest plot from a meta-analysis

**Description**

Generate a forest plot from a meta-analysis

**Usage**

```r
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(quote(paste(quote(100 * I^2)), 
    I^2 = rlang::quo(sprintf("%0.1f%%", I2)), 
    p = 
    rlang::quo(format.pval(QEp, digits = 4, eps = 1e-04, scientific = 1))), 
    trans = I, 
    func = 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`
additional_data

- a data.frame of additional data that can be referenced for the data shown in the panels of the forest plot

point_size

- 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

model_label

- a single model label or character vector of model labels the same length as model

show_individual_studies

- whether to show the individual studies (the default) or just the summary diamond

show_model

- a logical value, if `TRUE`, show model result, otherwise only show forest plots for studies

show_stats

- a list of stats to show at the bottom of the forest plot for e.g. heterogeneity

trans

- an optional transform function used on the numeric data for plotting the axes

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 `exponentiate == TRUE`)

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

Details

This produces a forest plot using the `rma`

Value

plot

Examples

```r
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)
  }
}
```
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_forest

<table>
<thead>
<tr>
<th>theme</th>
<th>theme to apply to the plot</th>
</tr>
</thead>
<tbody>
<tr>
<td>limits</td>
<td>limits of the forest plot on the X-axis (taken as the range of the data by default)</td>
</tr>
<tr>
<td>breaks</td>
<td>breaks to appear on the X-axis (note these will be exponentiated if exponentiate == TRUE)</td>
</tr>
<tr>
<td>recalculate_width</td>
<td>TRUE to recalculate panel widths using the current device or the desired plot width in inches</td>
</tr>
<tr>
<td>recalculate_height</td>
<td>TRUE to shrink text size using the current device or the desired plot height in inches</td>
</tr>
<tr>
<td>exclude_infinite_cis</td>
<td>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</td>
</tr>
</tbody>
</table>

Value

A ggplot ready for display or saving

---

**theme_forest**

*Default forest theme*

**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