Package ‘forestmodel’
April 25, 2018

Type Package
Title Forest Plots from Regression Models
Version 0.5.0
Date 2018-04-25
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 (>= 2.2.1)
Imports dplyr (>= 0.7.4), broom (>= 0.4.4), lazyeval (>= 0.1.10),
tibble (>= 1.4.2)
Suggests survival, metafor
RoxygenNote 6.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2018-04-25 07:53:06 UTC

R topics documented:

  default_forest_panels ........................................ 2
  forest_breaks .................................................. 2
  forest_model ................................................... 3
  forest_panels .................................................. 5
  forest_rma ..................................................... 6
  panel_forest_plot ............................................ 7
  theme_forest .................................................. 8

Index 9
**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

**Value**  
A vector with breaks ready to pass to `panel_forest_plot`
Produce a forest plot based on a regression model

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 including `colour` of the point estimate and error bars, shape of the point estimate, banded whether to show light grey bands behind alternate rows, `text_size` size of text in mm
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 function takes the model output from one of the common model functions in R (e.g. \texttt{lm}, \texttt{glm}, \texttt{coxph}).

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), level (the level of factors), 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 estimate.

Value

A \texttt{ggplot} ready for display or saving, or (with \texttt{return\_data} == \texttt{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\-\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 <- data_frame(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_panels**

Generate panels for forest plots

### Description

Generate panels for forest plots

### Usage

```r
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  
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_stats = list("I^2" =
~asprintf("%.1f%%", I2), p = ~format.pval(QEp, digits = 4, eps = 1e-04,
scientific = 1)), trans = I, funcs = NULL, format_options = list(colour
= "black", shape = 15, text_size = 5, banded = TRUE),
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_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 including `colour` of the point estimate and error bars, `shape` of the point estimate, `banded` whether to show light grey bands behind alternate rows, `text_size` size of text in mm
- **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)
`panel_forest_plot`  

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))
  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))
}
```

---

**Description**

Plot a forest plot with panels of text

**Usage**

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

forest_data: data.frame with the data needed for both the plot and text

mapping: mapping aesthetic created using `aes` or `aes_string`

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 including: `colour` of the point estimate and error bars, `shape` of the point estimate, `banded` whether to show light grey bands behind alternate rows, `text_size` size of text in mm

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`)

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

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, 8
aes_string, 8

coxph, 3, 4

default_forest_panels, 2

forest_breaks, 2
forest_model, 3, 5
forest_panels, 5
forest_rma, 5, 6

glm, 3, 4

lm, 3, 4

panel_forest_plot, 2, 4, 7

rma, 6, 7

theme_forest, 8

tidy, 4