

Package ‘metaviz’

July 22, 2025

Type Package

Title Forest Plots, Funnel Plots, and Visual Funnel Plot Inference for
Meta-Analysis

Version 0.3.1

Description A compilation of functions to create visually appealing and information-rich plots of meta-analytic data using 'ggplot2'. Currently allows to create forest plots, funnel plots, and many of their variants, such as rainforest plots, thick forest plots, additional evidence contour funnel plots, and sunset funnel plots. In addition, functionalities for visual inference with the funnel plot in the context of meta-analysis are provided.

License GPL-2

Encoding UTF-8

LazyData true

Imports ggplot2 (>= 3.1.0), nullabor (>= 0.3.5), dplyr (>= 0.7.8),
RColorBrewer (>= 1.1-2), metafor (>= 1.9-9), gridExtra (>=
2.2.1), ggpubr (>= 0.1.6)

Suggests Cairo (>= 1.5-9), knitr, rmarkdown

Depends R (>= 3.2.0)

RoxygenNote 7.1.0

VignetteBuilder knitr

URL <https://github.com/Mkossmeier/metaviz>

BugReports <https://github.com/Mkossmeier/metaviz/issues>

NeedsCompilation no

Author Michael Kossmeier [cre, aut],
Ulrich S. Tran [aut],
Martin Voracek [aut]

Maintainer Michael Kossmeier <michael.kossmeier@univie.ac.at>

Repository CRAN

Date/Publication 2020-04-09 09:10:08 UTC

Contents

metaviz-package	2
brainvol	3
exrehab	4
funnelinf	5
homeopath	8
mozart	9
viz_forest	10
viz_funnel	13
viz_rainforest	17
viz_sunset	21
viz_thickforest	24
Index	28

metaviz-package	<i>metaviz: Forest Plots, Funnel Plots, and Visual Funnel Plot Inference for Meta-Analysis</i>
-----------------	--

Description

The package `metaviz` is a collection of functions to create visually appealing and information-rich plots of meta-analytic data using `ggplot2`. Functions to create several variants of forest plots (`viz_forest`), funnel plots (`viz_funnel`, `viz_sunset`), and to conduct visual inference with funnel plots (`funnelinf`) are provided.

Forest plots (`viz_forest`)

Several different types and variants of forest plots can be created. This includes classic forest plots, subgroup forest plots, cumulative summary forest plots, and leave-one-out sensitivity forest plots. In addition, the function allows to individually label and color studies and to align tables with further study-level and summary-level information.

In addition to traditional forest plots, rainforest plots as well as thick forest plots can be used. Rainforest and thick forest plots are two recently proposed variants and enhancements of the classic forest plot. Both variants visually emphasize large studies (with short confidence intervals and more weight in the meta-analysis), while small studies (with wide confidence intervals and less weight in the meta-analysis) are visually less dominant. For further details see `help(viz_forest)`, `help(viz_rainforest)`, and `help(viz_thickforest)`.

Funnel plots (`viz_funnel`, `viz_sunset`)

Numerous different funnel plot variants can be created. Options for several graphical augmentations (e.g., confidence, significance, and additional evidence contours; choice of the ordinate; showing study subgroups), and different statistical information displayed are provided (Egger's regression line, and imputed studies by, as well as the adjusted summary effect from, the trim-and-fill method). Further details and references can be found in the corresponding help file (`help(viz_funnel)`).

Moreover, a novel variant of the funnel plot is introduced which displays the power of studies to detect an effect of interest (e.g., the meta-analytic summary effect) using a two-sided Wald test. This sunset (power-enhanced) funnel plot uses color-coded regions and a second y axis to visualize study-level power and can help to critically examine the evidentiality and credibility of a set of studies. For further details see `help(viz_sunset)`.

Visual inference with funnel plots (`funnelinf`)

Funnel plots are widely used in meta-analysis to assess small study effects as potential indicator of publication bias. Visual inference can help to improve the objectivity and validity of conclusions based on funnel plot examinations by guarding the meta-analyst from interpreting patterns in the funnel plot that might be perfectly plausible by chance. Only if the funnel plot showing the real data is distinguishable from simultaneously presented null funnel plots showing data simulated under the null hypothesis, conclusion based on visually inspecting the real-data funnel plot might be warranted. The function `funnelinf` provides numerous tailored options to conduct visual inference with the funnel plot graph in the context of meta-analysis. See `help(funnelinf)` for further details and relevant references.

Meta-analytic example datasets

Four different example datasets from published meta-analyses are distributed with the package:

- Two datasets for meta-analysis with standardized mean differences (`mozart`, `homeopath`)
- One dataset for meta-analysis with correlation coefficients (`brainvol`)
- One dataset for meta-analysis with dichotomous outcome data (`exrehab`).

More details and corresponding references can be found in the respective help files (`help(mozart)`, `help(homeopath)`, `help(brainvol)`, `help(exrehab)`).

brainvol

Example data for a meta-analysis of correlations: Human brain volume and intelligence

Description

A dataset consisting of 83 empirical studies used in the meta-analysis of Pietschnig, Penke, Wicherts, Zeiler, and Voracek (2015) on the associations between human brain volume and intelligence (measured full-scale IQ) in healthy participant samples.

Usage

brainvol

Format

A data frame with 83 rows and 8 variables:

study_name short name of each study

year publication year of each study

r observed correlations between intelligence and brain size

z Fisher's z transform of the observed correlations for meta-analysis

z_se Standard error of Fisher's z transformed observed correlations for meta-analysis

n sample size of each study

sex categorical moderator variable: gender of the participants in each study

mean_age continuous moderator variable: mean age of all participants in each study

References

Pietschnig, J., Penke, L., Wicherts, J. M., Zeiler, M., & Voracek, M. (2015). Meta-analysis of associations between human brain volume and intelligence differences: How strong are they and what do they mean?. *Neuroscience & Biobehavioral Reviews*, 57, 411-432.

exrehab

Example data for a meta-analysis of dichotomous outcomes: Exercise-based cardiac rehabilitation

Description

A dataset consisting of 14 empirical studies used for a meta-analysis in Anderson et al. (2016). The outcome of interest was risk of hospital admission of patients with coronary heart disease within follow up duration. Compared was exercise-based cardiac rehabilitation (treatment) with usual care (control).

Usage

exrehab

Format

A data frame with 14 rows and 15 variables:

study_name short name of each study

year publication year of each study

ai number of patients in the treatment group with an event (hospital admission) for each study.

bi number of patients in the treatment group with no event for each study.

ci number of patients in the control group with an event (hospital admission) for each study.

di number of patients in the control group with no event for each study.

nli number of patients in the treatment group for each study, (ai + bi).

- n2i** number of patients in the control group for each study, $(c_i + d_i)$.
- rr** relative risk of an event for treatment vs. control, $(a_i/n1_i)/(c_i/n2_i)$.
- or** odds ratio of an event for treatment vs. control, $(a_i*d_i)/(b_i*c_i)$.
- logrr** natural logarithm of the relative risk (rr) for meta-analysis.
- logrr_se** standard error of the natural logarithm of the relative risk for meta-analysis, $\sqrt{1/a_i + 1/c_i - 1/(a_i + b_i) - 1/(c_i + d_i)}$.
- logor** natural logarithm of the odds ratio (or) for meta-analysis.
- logor_se** standard error of the natural logarithm of the odds ratio for meta-analysis, $\sqrt{1/a_i + 1/b_i + 1/c_i + 1/d_i}$.
- followup** dichotomous moderator: follow up duration.

References

Anderson, L., Oldridge, N., Thompson, D. R., Zwisler, A. D., Rees, K., Martin, N., & Taylor, R. S. (2016). Exercise-based cardiac rehabilitation for coronary heart disease: Cochrane systematic review and meta-analysis. *Journal of the American College of Cardiology*, *67*, 1-12.

funnelinf

Visual funnel plot inference for meta-analysis

Description

Creates a lineup of funnel plots to conduct visual funnel plot inference (Kossmeier, Tran, & Voracek, 2019). The funnel plot showing the actually supplied data is presented alongside null plots showing simulated data under the null hypothesis.

Usage

```
funnelinf(
  x,
  group = NULL,
  group_permut = FALSE,
  n = 20,
  null_model = "REM",
  y_axis = "se",
  contours = TRUE,
  sig_contours = TRUE,
  contours_col = "Blues",
  trim_and_fill = FALSE,
  trim_and_fill_side = "left",
  egger = FALSE,
  show_solution = FALSE,
  rorschach = FALSE,
  point_size = 1.5,
  text_size = 3,
```

```

xlab = "Effect",
ylab = NULL,
x_trans_function = NULL,
x_breaks = NULL
)

```

Arguments

<code>x</code>	data.frame or matrix with the effect sizes of all studies (e.g., correlations, log odds ratios, or Cohen d) in the first column and their respective standard errors in the second column. Alternatively, <code>x</code> can be the output object of function <code>rma.uni</code> from package metafor ; then effect sizes and standard errors are extracted from <code>x</code> .
<code>group</code>	factor indicating the subgroup of each study to show in the funnel plot. Has to be in the same order than <code>x</code> .
<code>group_permut</code>	logical scalar indicating if subgroup membership should be permuted in the null plots. Ignored if no <code>group</code> is supplied.
<code>n</code>	integer specifying the absolute number of plots in the lineup.
<code>null_model</code>	character string indicating which meta-analytic model should be used to simulate the effect sizes for the null plots. Available options are "FEM" for the fixed effect model and "REM" (default) for the random-effects model (using the DerSimonian-Laird method to estimate the between-study variance τ^2).
<code>y_axis</code>	character string indicating which y axis should be used in the funnel plot. Available options are "se" (default) for standard error and "precision" for the reciprocal of the standard error.
<code>contours</code>	logical scalar indicating if classic funnel plot confidence contours and the summary effect should be displayed (i.e., summary effect +/- $qnorm(0.975) * SE$).
<code>sig_contours</code>	logical scalar. Should significance contours be drawn? Significance contours show which combination of effect size and standard error lead to study p-values smaller than 0.05 or 0.01 (using a Wald test).
<code>contours_col</code>	character string indicating the color palette used from package RColorBrewer for <code>sig_contours</code> . Can be any of "Blues", "Greys", "Oranges", "Greens", "Reds", and "Purples".
<code>trim_and_fill</code>	logical scalar. Should studies imputed by the trim and fill method be displayed? Also shows the adjusted summary effect if <code>contours</code> is TRUE as well.
<code>trim_and_fill_side</code>	character string indicating on which side of the funnel plot studies should be imputed by the trim and fill method (i.e., on which side studies are presumably missing due to publication bias). Must be either "right" or "left" (default).
<code>egger</code>	logical scalar. Should Egger's regression line be drawn? Only available if <code>y_axis</code> is "se".
<code>show_solution</code>	logical scalar. Should the real-data plot be highlighted?
<code>rorschach</code>	logical scalar. Should the lineup only consist of null plots?
<code>point_size</code>	numeric value. Size of the study points in the funnel plots.

text_size	numeric value. Size of text in the lineup.
xlab	character string specifying the label of the x axis.
ylab	character string specifying the label of the y axis.
x_trans_function	function to transform the labels of the x axis. Common uses are to transform log-odds-ratios or log-risk-ratios with <code>exp</code> to their original scale (odds ratios and risk ratios), or Fisher's z values back to correlation coefficients using <code>tanh</code> .
x_breaks	numeric vector of values for the breaks on the x-axis. When used in tandem with <code>x_trans_function</code> the supplied values should be not yet transformed.

Details

Funnel plots are widely used in meta-analysis to assess small study effects as potential indicator for publication bias. However, interpretations of funnel plots often lead to false conclusions about the presence and severity of bias (e.g., Terrin, Schmid, and Lau, 2005). Visual inference (Buja et al. 2009; Majumder, Hofmann, and Cook 2013) can help to improve the validity of conclusions based on the visual inspection of a funnel plot by saving investigators from interpreting funnel-plot patterns which might be perfectly plausible by chance. Only if the real-data funnel plot is distinguishable from null-plots, the null hypothesis is formally rejected and conclusions based on the visual inspection of the real-data funnel plot might be warranted (for further details, see Kossmeier, Tran, & Voracek, 2019).

Function `funnelinf` utilizes package **nullabor** for null plot simulation and **ggplot2** for plotting the lineup. Several tailored features for visual inference with funnel plots are provided which currently include:

1. options for null-plot simulation under both FEM and REM meta-analysis (see below).
2. subgroup analysis.
3. graphical options specific to the funnel plot (significance and confidence contours, and choice of the ordinate).
4. additional options to display various statistical information (Egger's regression line, and imputed studies by, as well as the adjusted summary effect from, the trim-and-fill method).

Null plots are simulated assuming normally distributed effect sizes with expected value equal to the observed summary effect and variance either equal to the observed study variances (`null_model = "FEM"`) or the sum of the observed study variances and the estimated between study variance τ^2 (`null_model = "REM"`).

Value

A lineup of `n` (20 by default) funnel plots; one showing the real data and `n-1` showing simulated data under the null hypothesis

Author(s)

Michael Kossmeier* <michael.kossmeier@univie.ac.at>

Ulrich S. Tran* <ulrich.tran@univie.ac.at>

Martin Voracek* <martin.voracek@univie.ac.at>

*Department of Basic Psychological Research and Research Methods, School of Psychology, University of Vienna

References

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E. K., Swayne, D. F., & Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367, 4361-4383.

Kossmeier, M., Tran, U. & Voracek, M. (2019) Visual inference for the funnel plot in meta-analysis. *Zeitschrift für Psychologie - Journal of Psychology*, 227.

Majumder, M., Hofmann, H., & Cook, D. (2013). Validation of visual statistical inference, applied to linear models. *Journal of the American Statistical Association*, 108, 942-956.

Terrin, N., Schmid, C. H., & Lau, J. (2005). In an empirical evaluation of the funnel plot, researchers could not visually identify publication bias. *Journal of clinical epidemiology*, 58, 894-901.

Examples

```
## Not run:
# Plotting a funnel plot lineup with the exrehab data to conduct visual funnel plot inference
funnelinf(x = exrehab[, c("logrr", "logrr_se")])

# Plotting a funnel plot lineup with the mozart data to conduct visual funnel plot inference
# considering subgroups
funnelinf(x = mozart[, c("d", "se")],
group = mozart[, "rr_lab"],
group_permut = TRUE, null_model = "REM")

# Plotting a funnel plot lineup with the brainvolume data to conduct visual funnel plot inference
# considering heterogeneity by using the fixed effect model for null plot simulation
funnelinf(x = brainvol[, c("z", "z_se")],
null_model = "FEM")

## End(Not run)
```

homeopath

*Example data for a meta-analysis of standardized mean differences:
Homeopathic treatment vs. Placebo*

Description

A dataset consisting of 54 randomized control trials (RCTs) from a published meta-analysis (Mathie et al. 2017) comparing homeopathic treatment with placebo. See reference for further details.

Usage

homeopath

Format

A data frame with 54 rows and 4 variables:

name short name of each RCT

year publication year of each RCT

d observed effect size (Standardized mean differences). Negative effect sizes indicate superiority of homeopathic treatment.

se standard error of observed effect size

References

Mathie, R. T., Ramparsad, N., Legg, L. A., Clausen, J., Moss, S., Davidson, J. R., ... McConnachie, A. (2017). Randomised, double-blind, placebo-controlled trials of non-individualised homeopathic treatment: Systematic review and meta-analysis. *Systematic Reviews*, 6, 63.

mozart	<i>Example data for a meta-analysis of standardized mean differences: Mozart effect</i>
--------	---

Description

A dataset consisting of 38 empirical studies used in the meta-analysis of Pietschnig, Voracek, and Formann (2010) on the Mozart effect. Each study compared the spatial task performance of participants after hearing the first movement "allegro con spirito" of the Mozart sonata for two pianos in D major (KV 448) with the spatial task performance of participants who were exposed to a non-musical stimulus or no stimulus at all.

Usage

mozart

Format

A data frame with 38 rows and 6 variables:

study_name short name of each study

n sample size of each study

d observed effect size (Cohen *d*). Positive values correspond to higher mean performance in the group hearing the Mozart sonata

se standard error of observed effect size

unpublished dichotomous moderator variable: Was the study unpublished?

rr_lab dichotomous moderator variable: Was the study conducted in the lab of authors Rauscher or Rideout?

References

Pietschnig, J., Voracek, M., & Formann, A. K. (2010). Mozart effect-Shmozart effect: A meta-analysis. *Intelligence*, 38, 314-323.

`viz_forest`*Forest plot variants for meta-analyses*

Description

Creates a rainforest, thick forest, or classic forest plot.

Usage

```
viz_forest(  
  x,  
  group = NULL,  
  type = "standard",  
  variant = "classic",  
  method = "FE",  
  study_labels = NULL,  
  summary_label = NULL,  
  confidence_level = 0.95,  
  col = "Blues",  
  summary_col = col,  
  text_size = 3,  
  xlab = "Effect",  
  x_limit = NULL,  
  x_trans_function = NULL,  
  x_breaks = NULL,  
  annotate_CI = FALSE,  
  study_table = NULL,  
  summary_table = NULL,  
  table_headers = NULL,  
  table_layout = NULL,  
  ...  
)
```

Arguments

- | | |
|--------------------|--|
| <code>x</code> | data.frame or matrix with the effect sizes of all studies (e.g., correlations, log odds ratios, or Cohen <i>d</i>) in the first column and their respective standard errors in the second column. Alternatively, <code>x</code> can be the output object of function <code>rma.uni</code> from package metafor ; then effect sizes and standard errors are extracted from <code>x</code> . |
| <code>group</code> | factor indicating the subgroup of each study to plot a subgroup forest plot. Has to be in the same order than <code>x</code> . |
| <code>type</code> | character string indicating the type of forest plot to be plotted. Can be "standard" (default), "study_only", "summary_only", "cumulative", or "sensitivity". See 'Details'. |

variant	character string indicating the forest plot variant that should be plotted. Can be "classic" (default) for a traditional forest plot, "rain" for rainforest plot, "thick" for a thick forest plot.
method	character string indicating which method should be used to compute the study weights and summary effect(s). Can be any method argument from function <code>rma.uni</code> of package metafor (e.g., "FE" for the fixed effect model, or "DL" for the random effects model using the DerSimonian-Laird method to estimate τ^2). If input <code>x</code> is an output object of function <code>rma.uni</code> from package metafor , then the method is extracted from <code>x</code> .
study_labels	a character vector with names/identifiers to annotate each study in the forest plot. Has to be in the same order than <code>x</code> . Ignored if <code>study_table</code> and/or <code>summary_table</code> is supplied.
summary_label	a character string specifying the name to annotate the summary effect. If a subgroup analysis is plotted, <code>summary_label</code> should be a character vector with a name for each subgroup summary effect, arranged in the order of the levels of group. Ignored if <code>study_table</code> and/or <code>summary_table</code> is supplied.
confidence_level	numeric value. The confidence level for the plotted confidence intervals.
col	character string specifying the main color for plotting study-level results. For <code>variant = "thick"</code> and <code>variant = "classic"</code> can be a vector of length <code>nrow(x)</code> with colors for each study-level result individually. For <code>variant = "rain"</code> must be one of the following palettes from package RColorBrewer : "Blues", "Greys", "Oranges", "Greens", "Reds", or "Purples".
summary_col	character string specifying the main color for plotting the summary effect(s). For <code>variant = "thick"</code> and <code>variant = "classic"</code> can be a vector with colors for each subgroup summary effect individually. For <code>variant = "rain"</code> with <code>type = "summary_only"</code> must be one of the following palettes from package RColorBrewer : "Blues", "Greys", "Oranges", "Greens", "Reds", or "Purples".
text_size	numeric value. Size of text in the forest plot. Default is 3.
xlab	character string specifying the label of the x axis. By default also used for the header of the aligned table if <code>annotate_CI</code> is TRUE.
x_limit	numeric vector of length 2 with the limits (minimum, maximum) of the x axis.
x_trans_function	function to transform the labels of the x axis. Common uses are to transform log-odds-ratios or log-risk-ratios with <code>exp</code> to their original scale (odds ratios and risk ratios), or Fisher's z values back to correlation coefficients using <code>tanh</code> .
x_breaks	numeric vector of values for the breaks on the x-axis. When used in tandem with <code>x_trans_function</code> the supplied values should be not yet transformed.
annotate_CI	logical scalar. Should the effect size and confidence interval values be shown as text in an aligned table on the right-hand side of the forest plot?
study_table	a data.frame with additional study-level variables which should be shown in an aligned table. Has to be in the same order than <code>x</code> .
summary_table	a data.frame with additional summary-level information shown in an aligned table. If <code>group</code> is supplied, <code>summary_table</code> must have a row for each subgroup summary effect, arranged in the order of the levels of group.

table_headers	character vector. Headers for each column of aligned tables via <code>study_table</code> , <code>summary_table</code> , and/or <code>annotate_CI</code> .
table_layout	numeric layout matrix passed to <code>layout_matrix</code> of <code>arrangeGrob</code> . Can be used to overwrite the default spacing of the forest plot and aligned tables via <code>study_table</code> , <code>summary_table</code> , and <code>annotate_CI</code> .
...	further arguments passed to <code>viz_rainforest</code> for <code>variant = "rain"</code> , or <code>viz_thickforest</code> for <code>variant = "thick"</code> .

Details

The forest plot is the most widely used display to visualize meta-analytic results. The function `viz_forest` creates visually appealing and informative-rich forest plots using `ggplot2`. Many options to flexibly customize the visual appearance and statistical information displayed are provided. In addition, rainforest plots as well as the thick forest plots can be created, two variants and enhancements of the classical forest plot recently proposed by Schild and Voracek (2015). For further details see the documentation of the wrapper functions `viz_rainforest`, and `viz_thickforest`.

Available forest plot types

Different aspects of meta-analytic data can be shown in forest plots. Five different types are available in `viz_forest` via the `type` parameter. Argument `"standard"` (default) shows study results as well as summary results in the forest plot. `"study_only"` allows to only show study results without the meta-analytic summary estimate. `"summary_only"` can be used to only show meta-analytic summary estimate(s), which is primarily useful to visualize several subgroup results (using `group`). `"cumulative"` shows a cumulative meta-analysis, that is, meta-analytic summary effects are computed sequentially by adding each study one-by-one. Studies are added in the same order than they were supplied in `x`. Finally, `"sensitivity"` shows for each study the meta-analytic summary effect if that particular study is not considered in the computation of the summary effect (leave-one-out analysis).

Value

A forest plot is created using `ggplot2`.

Author(s)

Michael Kossmeier* <michael.kossmeier@univie.ac.at>

Ulrich S. Tran* <ulrich.tran@univie.ac.at>

Martin Voracek* <martin.voracek@univie.ac.at>

*Department of Basic Psychological Research and Research Methods, School of Psychology, University of Vienna

References

Schild, A. H., & Voracek, M. (2015). Finding your way out of the forest without a trail of bread crumbs: Development and evaluation of two novel displays of forest plots. *Research Synthesis Methods*, 6, 74-86.

Examples

```

library(metaviz)
# Plotting the mozart data using a classic forest plot
viz_forest(x = mozart[, c("d", "se")],
study_labels = mozart[, "study_name"], xlab = "Cohen d")

# Subgroup analysis of published and unpublished studies shown in a rainforest plot
viz_forest(x = mozart[, c("d", "se")], study_labels = mozart[, "study_name"], method = "REML",
variant = "rain", summary_label = c("Summary (rr_lab = no)", "Summary (rr_lab = yes)"),
group = mozart[, "rr_lab"], xlab = "Cohen d")

# Thick forest plot with additional information in aligned tables. Log risk
# ratios are labeled in their original metric (risk ratios) on the x axis.
viz_forest(x = exrehab[, c("logrr", "logrr_se")], variant = "thick",
xlab = "RR", x_trans_function = exp, annotate_CI = TRUE,
study_table = data.frame(
Name = exrehab[, "study_name"],
eventsT = paste(exrehab$ai, "/", exrehab$ai + exrehab$bi, sep = ""),
eventsC = paste(exrehab$ci, "/", exrehab$ci + exrehab$di, sep = "")),
summary_table = data.frame(
Name = "Summary",
eventsT = paste(sum(exrehab$ai), "/", sum(exrehab$ai + exrehab$bi), sep = ""),
eventsC = paste(sum(exrehab$ci), "/", sum(exrehab$ci + exrehab$di), sep = "")),
table_layout = matrix(c(1, 1, 2, 2, 3), nrow = 1))

```

viz_funnel

*Funnel plot variants for meta-analysis***Description**

Creates a funnel plot. Many options regarding the appearance and statistical information displayed are provided (e.g., significance contours, additional evidence contours, and trim-and-fill analysis).

Usage

```

viz_funnel(
  x,
  group = NULL,
  y_axis = "se",
  method = "FE",
  contours = TRUE,
  sig_contours = TRUE,
  addev_contours = FALSE,
  contours_col = "Blues",
  contours_type = "FEM",
  detail_level = 1,
  egger = FALSE,
  trim_and_fill = FALSE,

```

```

trim_and_fill_side = "left",
text_size = 3,
point_size = 2,
xlab = "Effect",
ylab = NULL,
group_legend = FALSE,
group_legend_title = "",
x_trans_function = NULL,
x_breaks = NULL
)

```

Arguments

x	data.frame or matrix with the effect sizes of all studies (e.g., correlations, log odds ratios, or Cohen d) in the first column and their respective standard errors in the second column. Alternatively, x can be the output object of function <code>rma.uni</code> from package metafor ; then effect sizes and standard errors are extracted from x.
group	factor indicating the subgroup of each study to show in the funnel plot. Has to be in the same order than x.
y_axis	character string indicating which y axis should be used in the funnel plot. Available options are "se" (default) for standard error and "precision" for the reciprocal of the standard error.
method	character string indicating the method used to compute the meta-analytic summary effect and, for a random effects model, the between-study variance τ^2 . Can be any method argument from <code>rma.uni</code> (e.g., "FE" for the fixed effect model, or "DL" for the random effects model using the DerSimonian-Laird method to estimate τ^2). If input x is an output object of function <code>rma.uni</code> from package metafor , then the method is extracted from x.
contours	logical scalar indicating if classic funnel plot confidence contours and the summary effect should be displayed.
sig_contours	logical scalar. Should significance contours be drawn (at the 0.05 or 0.01 level using a Wald test)?
addev_contours	logical scalar. Should approximate additional evidence contours be drawn, showing the significance of the updated summary effect? See Details. Note: For method other than "FE" or "DL" runtime is increased significantly. Consider reducing <code>detail_level</code> .
contours_col	character string indicating the color palette used from package RColorBrewer for <code>sig_contours</code> , and <code>addev_contours</code> . Can be any of "Blues", "Greys", "Oranges", "Greens", "Reds", and "Purples".
contours_type	character string indicating how confidence contours are computed. Must be either "FEM" or "REM". If <code>contours_type</code> is set to "FEM" (default), contours are given as $M \pm qnorm(0.975) * SE$, with M the meta-analytic summary effect and SE the study standard error. If <code>contours_type</code> is set to "REM", contours are given as $M \pm qnorm(0.975) * \sqrt{SE^2 + \tau^2}$, with M again the meta-analytic summary effect, SE the study standard error and τ^2 the estimated between study

	heterogeneity from a random effects model. The argument of contours_type has no effect if τ^2 is zero.
detail_level	numeric scalar. Allows to increase or decrease the plotting detail of contours. Values can be chosen between 0.1 and 10. Default is 1.
egger	logical scalar. Should Egger's regression line be drawn? Only available if y_axis is "se".
trim_and_fill	logical scalar. Should studies imputed by the trim and fill method be displayed? Also shows the adjusted summary effect if contours is TRUE as well.
trim_and_fill_side	character string indicating on which side of the funnel plot studies should be imputed by the trim and fill method (i.e., on which side are studies presumably missing due to publication bias). Must be either "right" or "left" (default).
text_size	numeric value. Size of text in the funnel plot. Default is 3.
point_size	numeric value. Size of the study points in the funnel plot. Default is 2.
xlab	character string specifying the label of the x axis.
ylab	character string specifying the label of the y axis.
group_legend	logical scalar. Should there be a legend shown at the bottom of the graph if group was supplied?
group_legend_title	a character string specifying the title of the legend if group was supplied and group_legend is TRUE.
x_trans_function	function to transform the labels of the x axis. Common uses are to transform log-odds-ratios or log-risk-ratios with exp to their original scale (odds ratios and risk ratios), or Fisher's z values back to correlation coefficients using tanh.
x_breaks	numeric vector of values for the breaks on the x-axis. When used in tandem with x_trans_function the supplied values should be not yet transformed.

Details

The funnel plot is a widely used diagnostic plot in meta-analysis to assess small study effects and in particular publication bias. The function `viz_funnel` is capable to create a large set of different funnel plot variants. Options for several graphical augmentations (e.g., confidence, significance, and additional evidence contours; choice of the ordinate; study subgroups), and different statistical information displayed are provided (Egger's regression line, and imputed studies by, as well as the adjusted summary effect from, the trim-and-fill method).

Contours

Three different contours are available in `viz_funnel`:

1. **confidence contours** (argument `contours`) show the region where one expects 95% of all studies to fall (assuming the meta-analytic model applied is true and all estimates are identical to the parameters of interest). Confidence contours can help to assess the plausibility of observations given the meta-analytic model specified (fixed effect or random effects model).

2. **significance contours** (argument `sig_contours`) show shaded regions of individual study significance at the 5% and 1% level (using the standard errors supplied and a Wald test). Significance contours were proposed to help distinguish publication bias from other sources of funnel plot asymmetry (Peters, Sutton, Jones, Abrams, & Rushton, 2008).
3. **additional evidence contours: Significance of the summary effect** (argument `addev_contours`). These contours define regions where a new study has to fall such that the updated meta-analytic summary effect is significantly different from zero or not (using a two-sided test and an alpha level of 5%). Additional evidence contours allow to assess the robustness of the meta-analysis with respect to the effect of potentially new published evidence on the significance of the meta-analytic summary effect (Langan, Higgins, Gregory, & Sutton, 2012).

Measure on the y-axis

Two different options for the y-axis choice are available. First, to plot the standard errors on a reversed axis (i.e., studies with small standard errors are at the top). Second, precision (i.e., 1 divided by the standard error) can be used. Standard errors on the y-axis should be preferred in most situations but precision might have advantages if one or few large studies (with high precision) should be compared to the results of smaller studies condensed at the bottom of the funnel plot (Sterne & Egger, 2001).

Egger's regression line

Egger's regression line (Egger, Smith, Schneider & Minder, 1997) can be displayed if the standard error is used on the y axis. Classic Egger's regression can be computed as the OLS estimator of regressing the standardized effect size (effect size divided by its standard error) on precision (1 divided by the standard error). Showing this line in the funnel plot can further help to visually assess funnel plot asymmetry.

Trim and fill analysis

Imputed studies by the trim-and fill method, as well as the adjusted summary effect (Duval & Tweedie, 2000) can be displayed. The trim-and fill algorithm basically estimates the number of (extreme) studies responsible for funnel plot asymmetry. It then trims this number of (extreme) studies and computes the adjusted summary effect only considering the remaining studies. Finally, it imputes studies - presumably missing due to publication bias - by mirroring the trimmed (extreme) studies (driving the funnel plot asymmetry) around the (adjusted) summary effect. The user has to specify on which side of the funnel plot the trim-and fill method should impute missing studies (i.e., the direction were studies are presumably missing due to publication bias). To estimate the number of (extreme) studies responsible for funnel plot asymmetry the L estimator defined in Duval and Tweedie (2000) is used.

Value

A funnel plot is created using `ggplot2`.

Author(s)

Michael Kossmeier* <michael.kossmeier@univie.ac.at>

Ulrich S. Tran* <ulrich.tran@univie.ac.at>

Martin Voracek* <martin.voracek@univie.ac.at>

*Department of Basic Psychological Research and Research Methods, School of Psychology, University of Vienna

References

- Duval, S., & Tweedie, R. (2000). Trim and fill: a simple funnel-plot-based method of testing and adjusting for publication bias in meta-analysis. *Biometrics*, *56*, 455-463.
- Egger, M., Smith, G. D., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. *Bmj*, *315*, 629-634.
- Langan, D., Higgins, J. P., Gregory, W., & Sutton, A. J. (2012). Graphical augmentations to the funnel plot assess the impact of additional evidence on a meta-analysis. *Journal of clinical epidemiology*, *65*, 511-519.
- Peters, J. L., Sutton, A. J., Jones, D. R., Abrams, K. R., & Rushton, L. (2008). Contour-enhanced meta-analysis funnel plots help distinguish publication bias from other causes of asymmetry. *Journal of clinical epidemiology*, *61*, 991-996.
- Sterne, J. A., & Egger, M. (2001). Funnel plots for detecting bias in meta-analysis: guidelines on choice of axis. *Journal of clinical epidemiology*, *54*, 1046-1055

Examples

```
library(metaviz)
# Create a funnel plot using confidence and significance contours
viz_funnel(x = mozart[, c("d", "se")])

# Show a trim-and-fill analysis and Egger's regression line:
viz_funnel(x = mozart[, c("d", "se")], contours = TRUE,
trim_and_fill = TRUE, trim_and_fill_side = "left", egger = TRUE)

# Plot log-odds-ratios on the original OR scale and show additional evidence contours:
viz_funnel(x = exrehab[, c("logor", "logor_se")], sig_contours = FALSE,
addev_contours = TRUE, contours_col = "Greys", xlab = "Odds Ratio",
x_trans_function = exp, x_breaks = log(c(0.125, 0.25, 0.5, 1, 2, 4, 8)))

# Show study subgroups
viz_funnel(x = mozart[, c("d", "se")], group = mozart[, "unpublished"],
group_legend_title = "unpublished?")
```

viz_rainforest

Rainforest plots for meta-analyses

Description

Creates a rainforest plot, a novel variant of the forest plot.

Usage

```
viz_rainforest(
  x,
  group = NULL,
  type = "standard",
  method = "FE",
```

```

study_labels = NULL,
summary_label = NULL,
confidence_level = 0.95,
detail_level = 1,
col = "Blues",
summary_col = col,
text_size = 3,
xlab = "Effect",
x_limit = NULL,
x_trans_function = NULL,
x_breaks = NULL,
annotate_CI = FALSE,
study_table = NULL,
summary_table = NULL,
table_headers = NULL,
table_layout = NULL,
...
)

rainforest(
  x,
  group = NULL,
  type = "standard",
  method = "FE",
  study_labels = NULL,
  summary_label = NULL,
  confidence_level = 0.95,
  detail_level = 1,
  col = "Blues",
  summary_col = col,
  text_size = 3,
  xlab = "Effect",
  x_limit = NULL,
  x_trans_function = NULL,
  x_breaks = NULL,
  annotate_CI = FALSE,
  study_table = NULL,
  summary_table = NULL,
  table_headers = NULL,
  table_layout = NULL,
  ...
)

```

Arguments

x data.frame or matrix with the effect sizes of all studies (e.g., correlations, log odds ratios, or Cohen *d*) in the first column and their respective standard errors in the second column. Alternatively, **x** can be the output object of function `rma.uni` from package **metafor**; then effect sizes and standard errors are extracted from

	x.
group	factor indicating the subgroup of each study to plot a subgroup forest plot. Has to be in the same order than x.
type	character string indicating the type of forest plot to be plotted. Can be "standard" (default), "study_only", "summary_only", "cumulative", or "sensitivity". See 'Details'.
method	character string indicating which method should be used to compute the study weights and summary effect(s). Can be any method argument from rma.uni (e.g., "FE" for the fixed effect model, or "DL" for the random effects model using the DerSimonian-Laird method to estimate τ^2). If input x is an output object of function rma.uni from package metafor , then the method is extracted from x.
study_labels	a character vector with names/identifiers to annotate each study in the forest plot. Has to be in the same order than x. Ignored if <code>study_table</code> and/or <code>summary_table</code> is supplied.
summary_label	a character string specifying the name to annotate the summary effect. If a subgroup analysis is plotted, <code>summary_label</code> should be a character vector with a name for each subgroup summary effect, arranged in the order of the levels of group. Ignored if <code>study_table</code> and/or <code>summary_table</code> is supplied.
confidence_level	numeric value. The confidence level for the plotted confidence intervals and likelihood raindrops.
detail_level	numeric value. Values larger than 1 lead to a higher plotting detail (i.e., smoother likelihood raindrop polygons and more fluent color shading), values smaller than 1 to less plotting detail compared to the default plot.
col	character string specifying the color palette for plotting study-level results from package RColorBrewer . Can be any of "Blues", "Greys", "Oranges", "Greens", "Reds", and "Purples".
summary_col	character string specifying the color for plotting the summary effect(s). Can be any of "Blues", "Greys", "Oranges", "Greens", "Reds", and "Purples".
text_size	numeric value. Size of text in the forest plot. Default is 3.
xlab	character string specifying the label of the x axis. By default also used for the header of the aligned table if <code>annotate_CI</code> is TRUE.
x_limit	numeric vector of length 2 with the limits (minimum, maximum) of the x axis.
x_trans_function	function to transform the labels of the x axis. Common uses are to transform log-odds-ratios or log-risk-ratios with <code>exp</code> to their original scale (odds ratios and risk ratios), or Fisher's z values back to correlation coefficients using <code>tanh</code> .
x_breaks	numeric vector of values for the breaks on the x-axis. When used in tandem with <code>x_trans_function</code> the supplied values should be not yet transformed.
annotate_CI	logical scalar. Should the effect size and confidence interval values be shown as text in an aligned table on the right-hand side of the forest plot?
study_table	a data.frame with additional study-level variables which should be shown in an aligned table. Has to be in the same order than x.

summary_table	a data.frame with additional summary-level information shown in an aligned table. If group is supplied, summary_table must have a row for each subgroup summary effect, arranged in the order of the levels of group.
table_headers	character vector. Headers for each column of aligned tables via study_table, summary_table, or annotate_CI.
table_layout	numeric layout matrix passed to layout_matrx of <code>arrangeGrob</code> . Can be used to overwrite the default spacing of the forest plot and aligned tables via study_table, summary_table, and annotate_CI.
...	deprecated argument names from earlier versions can still be passed to viz_rainforest via ...

Details

Rainforest plots were proposed by Schild and Voracek (2015) as a variant and enhancement of classic forest plots. Rainforest plots use (log-)likelihood drops to depict study level results (for details, see Barrowman & Myers, 2003). `viz_rainforest` assumes normality of effect sizes to construct these (log-)likelihood drops. The width of each such raindrop is identical to the width of the confidence interval. For a given (log-)likelihood raindrop, the height can be interpreted as the plausibility (i.e., (log-)likelihood value) for different true values given the observed estimate. Moreover, the height of each raindrop is scaled with respect to its relative meta-analytic weight considering all studies. Therefore, visually comparing heights of different raindrops highlights the relative importance within the meta-analysis. In addition, color shading is utilized to further visualize statistical uncertainty, as suggested by Jackson (2008). Finally, study and summary level point estimates are depicted clearly by a specific symbol.

Rainforest plots have the following advantages, as compared to classic forest plots:

1. The width of the likelihood raindrops corresponds to the confidence intervals, as also shown in the classic forest plot. In addition, for each likelihood drop the height (and color shading) visualizes the plausibility of true values given the observed estimate.
2. Low likelihood drops and light color shading causes small studies (with wide confidence intervals and less weight in the meta-analysis) to be visually less dominant.
3. In classic forest plots, it is often hard to depict the magnitude of point estimates to a reasonable degree of accuracy, especially for studies with large meta-analytic weights and correspondingly large plotting symbols (commonly squares). Specific symbols within the raindrops improve the visualization of study point estimates.

Note that for subgroup analysis the height of each raindrop is scaled by the weight of each study within the subgroup divided by the total weight sum of all studies irrespective of subgroup. Therefore, with subgroups present, the overall impression of raindrop heights and color shading within a given subgroup compared to other subgroups conveys information about the relative precision of the meta-analytic subgroup estimates.

Value

A Rainforest plot is created using `ggplot2`.

Author(s)

Michael Kossmeier* <michael.kossmeier@univie.ac.at>

Ulrich S. Tran* <ulrich.tran@univie.ac.at>

Martin Voracek* <martin.voracek@univie.ac.at>

*Department of Basic Psychological Research and Research Methods, School of Psychology, University of Vienna

References

Barrowman, N. J., & Myers, R. A. (2003). Raindrop plots: A new way to display collections of likelihoods and distributions. *American Statistician*, *57*, 268-274.

Jackson, C. H. (2008). Displaying uncertainty with shading. *American Statistician*, *62*, 340-347.

Schild, A. H., & Voracek, M. (2015). Finding your way out of the forest without a trail of bread crumbs: Development and evaluation of two novel displays of forest plots. *Research Synthesis Methods*, *6*, 74-86.

Examples

```
library(metaviz)
# Plotting a rainforest plot using the mozart data
viz_rainforest(x = mozart[, c("d", "se")],
study_labels = mozart[, "study_name"], xlab = "Cohen d")

# Visualizing a subgroup analysis of published and unpublished studies
viz_rainforest(x = mozart[, c("d", "se")], group = mozart[, "rr_lab"],
study_labels = mozart[, "study_name"], method = "REML",
summary_label = c("Summary (rr_lab = no)", "Summary (rr_lab = yes)"),
xlab = "Cohen d")

# Showing additional information in aligned tables. Log risk ratios are labeled
# in their original metric (risk ratios) on the x axis.
viz_rainforest(x = exrehab[, c("logrr", "logrr_se")],
annotate_CI = TRUE, xlab = "RR", x_trans_function = exp,
study_table = data.frame(
Name = exrehab[, "study_name"],
eventsT = paste(exrehab$ai, "/", exrehab$ai + exrehab$bi, sep = ""),
eventsC = paste(exrehab$ci, "/", exrehab$ci + exrehab$di, sep = "")),
summary_table = data.frame(
Name = "Summary",
eventsT = paste(sum(exrehab$ai), "/", sum(exrehab$ai + exrehab$bi), sep = ""),
eventsC = paste(sum(exrehab$ci), "/", sum(exrehab$ci + exrehab$di), sep = "")))
```

viz_sunset

Sunset (power-enhanced) funnel plot

Description

Creates a funnel plot with power regions and computes power-related statistics.

Usage

```

viz_sunset(
  x,
  y_axis = "se",
  true_effect = NULL,
  method = "FE",
  sig_level = 0.05,
  power_stats = TRUE,
  power_contours = "discrete",
  contours = FALSE,
  sig_contours = TRUE,
  text_size = 3,
  point_size = 2,
  xlab = "Effect",
  ylab = NULL,
  x_trans_function = NULL,
  x_breaks = NULL,
  y_breaks = NULL,
  x_limit = NULL,
  y_limit = NULL
)

```

Arguments

x	data.frame or matrix with the effect sizes of all studies (e.g., log odds ratios, or Cohen d) in the first column and their respective standard errors in the second column. Alternatively, x can be the output object of function <code>rma.uni</code> from package metafor ; then effect sizes and standard errors are extracted from x.
y_axis	character string indicating which y axis should be used in the funnel plot. Available options are "se" (default) for standard error and "precision" for the reciprocal of the standard error.
true_effect	numeric scalar. Which true effect should be assumed for power calculations? The default is NULL, for which the meta-analytic summary effect is used (computed with method).
method	character string indicating the method used to compute the meta-analytic summary effect. Can be any method argument from <code>rma.uni</code> (e.g., "FE" for the fixed effect model (default), or "DL" for the random effects model using the DerSimonian-Laird method to estimate τ^2). If input x is an output object of function <code>rma.uni</code> from package metafor , then the method is extracted from x.
sig_level	logical scalar. For which significance level alpha should the study power be computed?
power_stats	logical scalar. Should power-related statistics be computed and printed in the caption of the plot? (see details)
power_contours	character string specifying how different power regions are plotted. Can be either "continuous" or "discrete" (default).
contours	logical scalar indicating if classic funnel plot confidence contours and the summary effect should be displayed.

sig_contours	logical scalar. Should significance contours be drawn (at the 0.05 or 0.01 level using a two-sided Wald test)?
text_size	numeric value. Size of text in the funnel plot. Default is 3.
point_size	numeric value. Size of the study points in the funnel plot. Default is 2.
xlab	character string specifying the label of the x axis.
ylab	character string specifying the label of the y axis.
x_trans_function	function to transform the labels of the x axis. Common uses are to transform log-odds-ratios or log-risk-ratios with <code>exp</code> to their original scale (odds ratios and risk ratios), or Fisher's z values back to correlation coefficients using <code>tanh</code> .
x_breaks	numeric vector of values for the breaks on the x-axis. When used in tandem with <code>x_trans_function</code> the supplied values should be not yet transformed.
y_breaks	numeric vector of values for the breaks on the y-axis.
x_limit	numeric vector of length two with user specified x axis limits.
y_limit	numeric vector of length two with user specified y axis limits.

Details

The funnel plot is the most widely used diagnostic plot in meta-analysis, primarily to assess small-study effects. The sunset (power-enhanced) funnel plot incorporates study-level power information in the funnel display. This directly allows to examine the power studies had to detect an effect of interest (e.g., the observed meta-analytic summary effect), whether funnel plot asymmetry is driven by underpowered but significant studies, and to visually assess if there is an excess of low-powered significant effects in the meta-analysis (conceptually related to the test of excess significance, Ioannidis & Trikalinos, 2007). For effect sizes assumed to be normally distributed (e.g., Cohen d, log OR), the power corresponding to a given standard error is computed by using a two-sided Wald test and (by default) the meta-analytic summary effect as assumed true effect. Colored regions of different power levels and a second axis with study level power are shown in the funnel plot. In addition, power-related statistics are shown: a) The median power of all studies, b) the true effect size necessary such that the median power of the studies would have been 33% or 66%, c) results of a test of excess significance (Ioannidis & Trikalinos, 2007), and d) the R-Index for expected replicability (Schimmack, 2016).

Value

A power enhanced ("sunset") funnel plot is created using `ggplot2`.

Author(s)

Michael Kossmeier* <michael.kossmeier@univie.ac.at>

Ulrich S. Tran* <ulrich.tran@univie.ac.at>

Martin Voracek* <martin.voracek@univie.ac.at>

*Department of Basic Psychological Research and Research Methods, School of Psychology, University of Vienna

References

Ioannidis, J. P., & Trikalinos, T. A. (2007). An exploratory test for an excess of significant findings. *Clinical Trials*, 4, 245-253.

Schimmack, U. (2016). The replicability-index: Quantifying statistical research integrity. Retrieved from <https://replicationindex.wordpress.com/2016/01/31/a-revised-introduction-to-the-r-index/>

Examples

```
library(metaviz)
# Create a power-enhanced ("sunset") funnel plot using confidence and significance contours
viz_sunset(x = homeopath[, c("d", "se")], contours = TRUE)
```

viz_thickforest

Thick forest plots for meta-analyses

Description

Creates a thick forest plot, a novel variant of the forest plot.

Usage

```
viz_thickforest(
  x,
  group = NULL,
  type = "standard",
  method = "FE",
  study_labels = NULL,
  summary_label = NULL,
  confidence_level = 0.95,
  col = "Blues",
  summary_col = col,
  tick_col = "firebrick",
  text_size = 3,
  xlab = "Effect",
  x_limit = NULL,
  x_trans_function = NULL,
  x_breaks = NULL,
  annotate_CI = FALSE,
  study_table = NULL,
  summary_table = NULL,
  table_headers = NULL,
  table_layout = NULL
)
```

Arguments

x	data.frame or matrix with the effect sizes of all studies (e.g., correlations, log odds ratios, or Cohen d) in the first column and their respective standard errors in the second column. Alternatively, x can be the output object of function <code>rma.uni</code> from package metafor ; then effect sizes and standard errors are extracted from x.
group	factor indicating the subgroup of each study to plot a subgroup forest plot. Has to be in the same order than x.
type	character string indicating the type of forest plot to be plotted. Can be "standard" (default), "study_only", "summary_only", "cumulative", or "sensitivity". See 'Details'.
method	character string indicating which method should be used to compute the study weights and summary effect(s). Can be any method argument from <code>rma.uni</code> (e.g., "FE" for the fixed effect model, or "DL" for the random effects model using the DerSimonian-Laird method to estimate τ^2). If input x is an output object of function <code>rma.uni</code> from package metafor , then the method is extracted from x.
study_labels	a character vector with names/identifiers to annotate each study in the forest plot. Has to be in the same order than x. Ignored if <code>study_table</code> and/or <code>summary_table</code> is supplied.
summary_label	a character string specifying the name to annotate the summary effect. If a subgroup analysis is plotted, <code>summary_label</code> should be a character vector with a name for each subgroup summary effect, arranged in the order of the levels of group. Ignored if <code>study_table</code> and/or <code>summary_table</code> is supplied.
confidence_level	numeric value. The confidence level for the plotted confidence bars.
col	character string specifying the color used for the study-level error bars. Can be a vector of length <code>nrow(x)</code> with colors for each study-level result individually.
summary_col	character string specifying the main color for plotting the summary effect(s). Can be a vector with colors for each subgroup summary effect individually.
tick_col	character string specifying the color used for the ticks indicating the point estimates.
text_size	numeric value. Size of text in the forest plot. Default is 3.
xlab	character string specifying the label of the x axis. By default also used for the header of the aligned table if <code>annotate_CI</code> is TRUE.
x_limit	numeric vector of length 2 with the limits (minimum, maximum) of the x axis.
x_trans_function	function to transform the labels of the x axis. Common uses are to transform log-odds-ratios or log-risk-ratios with <code>exp</code> to their original scale (odds ratios and risk ratios), or Fisher's z values back to correlation coefficients using <code>tanh</code> .
x_breaks	numeric vector of values for the breaks on the x-axis. When used in tandem with <code>x_trans_function</code> the supplied values should be not yet transformed.
annotate_CI	logical scalar. Should the effect size and confidence interval values be shown as text in an aligned table on the right-hand side of the forest plot?

study_table	a data.frame with additional study-level variables which should be shown in an aligned table. Has to be in the same order than x.
summary_table	a data.frame with additional summary-level information shown in an aligned table. If group is supplied, summary_table must have a row for each subgroup summary effect, arranged in the order of the levels of group.
table_headers	character vector. Headers for each column of aligned tables via study_table, summary_table, or annotate_CI.
table_layout	numeric layout matrix passed to layout_matrix of <code>arrangeGrob</code> . Can be used to overwrite the default spacing of the forest plot and aligned tables via study_table, summary_table, and annotate_CI.

Details

The thick forest plot was proposed by Schild and Voracek (2015) as a variant and enhancement of classic forest plots. Thick forest plots use rectangular error bars instead of traditional lines to display confidence intervals (width of the error bar), as well as the relative meta-analytic weight (height of the error bar) of each study. In addition, study and summary level point estimates are depicted clearly by a specific symbol.

Thick forest plots have the following advantages, as compared to classic forest plots:

1. Using the height of bars proportional to the (relative) meta-analytic weight causes small studies (with wide confidence intervals and less weight in the meta-analysis) to be visually less dominant.
2. In classic forest plots, it is often hard to depict the magnitude of point estimates to a reasonable degree of accuracy, especially for studies with large meta-analytic weights and correspondingly large plotting symbols (commonly squares). Specific symbols within the thick forest plot improve the visualization of study point estimates.

Note that for subgroup analysis the height of each error bar is scaled by the weight of each study within the subgroup divided by the sum of the weights of all studies irrespective of subgroup. Therefore, with subgroups present, the overall impression of error bar heights within a given subgroup compared to other subgroups conveys information about the relative precision of the meta-analytic estimate within the subgroup.

Value

A thick forest plot is created using `ggplot2`.

Author(s)

Michael Kossmeier* <michael.kossmeier@univie.ac.at>

Ulrich S. Tran* <ulrich.tran@univie.ac.at>

Martin Voracek* <martin.voracek@univie.ac.at>

*Department of Basic Psychological Research and Research Methods, School of Psychology, University of Vienna

References

Schild, A. H., & Voracek, M. (2015). Finding your way out of the forest without a trail of bread crumbs: Development and evaluation of two novel displays of forest plots. *Research Synthesis Methods*, 6, 74-86.

Examples

```
library(metaviz)
# Plotting a thick forest plot using the mozart data
viz_thickforest(x = mozart[, c("d", "se")],
  study_labels = mozart[, "study_name"], xlab = "Cohen d")

# Visualizing a subgroup analysis of published and unpublished studies
viz_thickforest(x = mozart[, c("d", "se")], group = mozart[, "rr_lab"],
  study_labels = mozart[, "study_name"], method = "REML",
  summary_label = c("Summary (rr_lab = no)", "Summary (rr_lab = yes)"),
  xlab = "Cohen d")

# Showing additional information in aligned tables. Log risk ratios are labeled
# in their original metric (risk ratios) on the x axis.
viz_thickforest(x = exrehab[, c("logrr", "logrr_se")],
  annotate_CI = TRUE, xlab = "RR", x_trans_function = exp,
  study_table = data.frame(
    Name = exrehab[, "study_name"],
    eventsT = paste(exrehab$ai, "/", exrehab$ai + exrehab$bi, sep = ""),
    eventsC = paste(exrehab$ci, "/", exrehab$ci + exrehab$di, sep = "")),
  summary_table = data.frame(
    Name = "Summary",
    eventsT = paste(sum(exrehab$ai), "/", sum(exrehab$ai + exrehab$bi), sep = ""),
    eventsC = paste(sum(exrehab$ci), "/", sum(exrehab$ci + exrehab$di), sep = "")),
  table_layout = matrix(c(1, 1, 2, 2, 3), nrow = 1))
```

Index

* datasets

- brainvol, [3](#)
- exrehab, [4](#)
- homeopath, [8](#)
- mozart, [9](#)

arrangeGrob, [12](#), [20](#), [26](#)

brainvol, [3](#)

exrehab, [4](#)

funnelinf, [5](#)

homeopath, [8](#)

metaviz (metaviz-package), [2](#)

metaviz-package, [2](#)

mozart, [9](#)

rainforest (viz_rainforest), [17](#)

rma.uni, [10](#), [11](#), [14](#), [18](#), [19](#), [22](#), [25](#)

viz_forest, [10](#)

viz_funnel, [13](#)

viz_rainforest, [12](#), [17](#)

viz_sunset, [21](#)

viz_thickforest, [12](#), [24](#)