Package ‘eclipseplot’

May 8, 2026

Title Graphical Visualizations for ROBUST-RCT Risk of Bias Assessments

Version 0.9.6

Description Provides visual representations of risk-of-bias assessments using the ROBUST-RCT framework, as described in Wang et al. (2025) <doi:10.1136/bmj-2024-081199>. The graphical visualization displays both factual evaluation (Step 1) and judgment (Step 2).

License GPL-3

Encoding UTF-8

LazyData true

URL `https://github.com/pevid-sci/eclipseplot`

BugReports `https://github.com/pevid-sci/eclipseplot/issues`

Imports ggplot2, dplyr, ggforce, cowplot, magrittr, tidyr, readxl

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

NeedsCompilation no

Author Pedro Rodrigues Vidor [aut, cre],
Yohan Casiraghi [ctb],
Sofia Simoni Rossi Fermo [ctb],
Adolfo Moraes de Souza [ctb],
Patricia Klarmann Ziegelmann [ctb],
Maria Inês Schmidt [ctb],
Maicon Falavigna [ctb]

Maintainer Pedro Rodrigues Vidor <`pedro.vidor@ufrgs.br`>

Repository CRAN

Date/Publication 2026-04-13 05:30:02 UTC

Contents

`eclipsedf` *Data Cleanning Helper for Eclipse Plot*

Description

Standardizes both column names and cell values to ensure compatibility with the 'eclipseplot' engine. It uses fuzzy matching via regular expressions to identify headers and a robust dictionary to map various shorthand judgments into the four standard risk-of-bias levels.

Usage

```
eclipsedf(df)
```

Arguments

`df`	A data frame to be cleaned.

Details

Column Name Standardization: The function identifies columns regardless of case and replaces them with:

- `study`: Matches "study", "article", "label", "paper", or "rct".
- `pmid`: Matches "pmid", "pm_id", "pm id", "article id", "article_id", or "id".
- `itemX_stepY`: Matches any string containing "item" followed by a number and "step" followed by a number (e.g., "Item 1, Step 2" becomes "item1_step2").

Value Standardization (Fuzzy Mapping): Judgment inputs are converted to lowercase and trimmed of whitespace. The mapping logic is as follows:

- **"Definitely Low"**: Result for "definitely yes", "definitely low", "dy", "dl", "yes", or "y".
- **"Probably Low"**: Result for "probably yes", "probably low", "py", or "pl".
- **"Probably High"**: Result for "probably no", "probably high", "pn", "ph", or "probably-no".
- **"Definitely High"**: Result for "definitely no", "definitely high", "dn", "dh", "no", or "n".
- **"Not applicable"**: Assigned to any other value, including empty cells (NA) or unrecognized text.

Value

A standardized data frame with specific categorical levels ("Definitely Low", "Probably Low", "Probably High", and "Definitely High") and column names (itemX_stepY) for use in the eclipse plot.

Note

For the purposes of the 'eclipseplot' function, there is no distinction between the internal mapping of Step 1 and Step 2 values. Both are standardized using the Step 2 visual scheme (color palette) to ensure a cohesive graphical representation, as the underlying directional logic remains the same.

#' @example data(messy) view(messy) organized <- eclipsedf(messy) view(organized)

`eclipseplot` *Create a plot for a risk of bias assessment with the ROBUST-RCT tool*

Description

A visual representation of risk-of-bias assessments of randomized controlled trials using the ROBUST-RCT framework, as described in Wang et al. (2025) <doi:10.1136/bmj-2024-081199>. It can display either the standard two-step assessment or a simplified judgment-only version.

Usage

```
eclipseplot(
  robust_data,
  standard = TRUE,
  optionals = FALSE,
  title = NULL,
  plot = "full",
  design = "horizontal",
  proportions = NULL,
  opt_prefix = "o"
)
```

Arguments

`robust_data`	A data frame containing the data from the risk-of-bias assessment with the ROBUST-RCT tool.
`standard`	Logical. If `TRUE` (default), displays the two-step eclipseplot. If `TRUE` (default), displays only the step 2 (judgment).
`optionals`	Logical. If `TRUE`, includes columns `opt1` to `opt9`.
`title`	Character. Custom title for the eclipse plot.
`plot`	Character. Parts to display: `"full"` (default), `"eclipses"`, or `"legend"`.
`design`	Character. Orientation of the assembly: `"horizontal"` (default) or `"vertical"`.
`proportions`	Numeric vector of length 2. Relative widths/heights for plot and legend.
`opt_prefix`	Character. Text for optional items in the plot. Default is "o".

Details

The input data frame must include the required columns for the ROBUST-RCT domains. The following columns are expected:

- **pmid:** PubMed ID or unique identifier.
- **study:** Study name (e.g., "Author, 2026").
- **Core Items (1-6):** Columns named 'itemX_step1' and 'itemX_step2' (where X is 1 to 6). As standard in the ROBUST-RCT tool, item 6 does not have a multiple-choice step 1 evaluation, so it is automatically marked as NA.
- **Optional Items:** If optionals = TRUE, columns named 'optX_step1' and 'optX_step2' (where X is 1 to 9).

Allowed values for Step 1 (Evaluation): "Definitely Yes", "Probably Yes", "Probably No", "Definitely No", "Not applicable".

Allowed values for Step 2 (Judgment): "Definitely Low", "Probably Low", "Probably High", "Definitely High", "Not applicable".

Value

A `ggplot` object (specifically a `ggdraw` object from the `cowplot` package) representing the visualization of the risk of bias. The output can be further customized using standard `ggplot2` functions or saved via `save_eclipseplot`.

Dataset Structure

Your CSV should look like this (Top 2 rows):

```
"pmid","study","item1_step1","item1_step2","item2_step1"..."item6_step2"
"40001","Albert, 2024","Definitely Yes","Definitely Low",..."Probably Low"
```

Color Mapping:

- Definitely Low / Definitely Yes: Blue (#4C72B0)
- Probably Low / Probably Yes: Light Blue (#c3d0e4)
- Probably High / Probably No: Light Red (#f5c7c7)
- Definitely High / Definitely No: Red (#E15759)
- Not applicable: Grey (#D3D3D3)

Author(s)

Pedro Rodrigues Vidor <`pedro.vidor@ufrgs.br`>

Yohan Casiraghi (Contributor)

Sofia Simoni Rossi Fermo (Contributor)

Adolfo Moraes de Souza (Contributor)

Patricia Klarmann Ziegelmann (Contributor)

Maria Inês Schmidt (Contributor)

Maicon Falavigna (Contributor)

Examples

```
# 1. Standard plot (Core items, Steps 1 & 2)
data(sample_brief)
eclipseplot_plot <- eclipseplot(sample_brief)
print(eclipseplot_plot)


# 2. Judgment only (Core items, Step 2)
eclipseplot(sample_long, standard = FALSE)

# 3. Plot with optional items (Core and optional items, Steps 1 & 2)
eclipseplot(sample_long, optionals = TRUE)

# 4. Judgment only with optional items (Core and optional items, Step 2)
eclipseplot(sample_long, standard = FALSE, optionals = TRUE)

# 5. Vertical full-page plot
eclipseplot(sample_long, design = "vertical", proportions = c(0.7, 0.3))

# 6. Plotting only the dots
eclipseplot(sample_brief, plot = "eclipses")

# 7. Plotting only the legend
eclipseplot(sample_brief, plot = "legend")
```

`from_xlsx` *Convert ROBUST Standard Spreadsheet to eclipseplot format*

Description

This function facilitates the conversion of data extracted using the standard ROBUST spreadsheet into the format required by 'eclipseplot'. It automatically identifies header locations, maps Step 1 and Step 2 columns for each domain, and filters out common spreadsheet artifacts (e.g., merged cells or labels like "Intervention Group").

Usage

```
from_xlsx(path, sheet = 1)
```

Arguments

`path`	Path to the '.xlsx' file.
`sheet`	Sheet name or index (defaults to 1).

Value

A standardized data frame formatted for 'eclipseplot' with specific column names (itemX_stepY) and categorical levels for bias assessment.

`get_eclipse_data` *Get path to example files (.csv)*

Description

Get path to example files (.csv)

Usage

```
get_eclipse_data(name = "brief")
```

Arguments

`name`	Name of the file ("brief" or "long").

Value

A character string containing the absolute file path to the requested sample CSV file within the package installation directory.

`messy` *Mock messy data for eclipsedf testing*

Description

A dataset containing messy headers and non-standard values to test the cleaning and standardization capabilities of the eclipsedf function.

Usage

```
messy
```

Format

A data frame with 4 rows and 13 variables of varied naming conventions.

`sample_brief` *Sample brief dataset for eclipseplot*

Description

A small dataset containing risk of bias assessments for testing.

Usage

```
sample_brief
```

Format

A data frame with 5 rows and 13 variables:

pmid PubMed ID

study Study citation label

item1_step1, item1_step2, item2_step1, item2_step2, item3_step1, item3_step2, item4_step1, item4_step2, item5_step
Risk of bias for the core items

`sample_long` *Sample long dataset for eclipseplot*

Description

A larger dataset with more studies and optional items.

Usage

```
sample_long
```

Format

A data frame with 26 rows and 13 variables.

`save_eclipseplot` *Export the eclipseplot to a file*

Description

Export the eclipseplot to a file

Usage

```
save_eclipseplot(plot_object, filename, width = 14, height = 9.8, dpi = 72)
```

Arguments

`plot_object`	The ggplot object returned by eclipseplot().
`filename`	Character. The name of the file to save. Must include extension (e.g., "plot.png").
`width`	Width in inches (Default is 14).
`height`	Height in inches (Default is 9.8).
`dpi`	Resolution (Default is 72).

Value

No return value, called for side effects (saves a file to the specified location).

Examples

```
# Using a temporary directory
data(sample_brief)
p <- eclipseplot(sample_brief)
tmp_file <- file.path(tempdir(), "test_plot.png")
save_eclipseplot(p, filename = tmp_file)
```

Index