Package 'eclipseplot'

November 2, 2026

Title Eclipse Plot: A Graphical Visualization for Risk of Bias Assessments with ROBUST

Version 1.0.0

Description Provides a novel visualization tool for the ROBUST framework, displaying both factual evaluation (Step 1) and judgment (Step 2).

License GPL-3

Encoding UTF-8

LazyData true

Imports ggplot2, dplyr, ggforce, cowplot, tidyr, readxl, grid, grDevices, tools, utils

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], Maicon Falavigna [ctb], Patricia Klarmann Ziegelmann [ctb]

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

Repository CRAN

Date/Publication 2026-11-02 13:50:17 UTC

Contents

check_eclipse_data 2
eclipsedf 3
eclipseplot 4
from_xlsx 7
get_eclipse_data 8
messy 8

read_eclipse 9
sample_brief 9
sample_long 10
save_eclipseplot 10

Index **12**

`check_eclipse_data` *Check a data set before plotting*

Description

Lists every problem that would make the Eclipse plot misleading: missing columns, empty or unrecognised ratings, duplicated study labels, optional columns that cannot be read, and (as information only) items 1 and 2 whose Step 2 rating is not the equivalent of the Step 1 rating.

Usage

```
check_eclipse_data(robust_data, standard = TRUE, optionals = TRUE)
```

Arguments

`robust_data`	A data frame in the eclipseplot format.
`standard`	Logical. If `TRUE`, Step 1 columns are required.
`optionals`	Logical. If `TRUE`, optional columns are checked.

Details

Ratings are compared without regard to case, extra spaces or hyphens, and the abbreviations accepted by `eclipsedf` (e.g. "py", "DL") are valid.

Value

A data frame with one row per problem and the columns `severity` ("error", "warning" or "info"), `row`, `study`, `column`, `value` and `message`. It has zero rows when no problem is found. Rows are numbered as in the input (row 1 = first study).

Examples

```
check_eclipse_data(sample_brief)
messy_values <- sample_brief
messy_values$item1_step1[1] <- "definately yes"
messy_values$item2_step2[2] <- NA
check_eclipse_data(messy_values)
```

`eclipsedf` *Data Cleaning 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:

- `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").
- `optX`: Matches "opt1", "Opt 2", "optional 3", "opt4_step2", ...
- `pmid`: Matches "pmid", "pm_id", "pm id", "article id", "study id", "trial id", or "id".
- `study`: Matches "study", "article", "label", "paper", "rct", "trial", "author", "reference" or "citation".

Identifier rules are applied before label rules, so "Article ID" becomes `pmid` and "Article" becomes `study`. If two columns match the same name, the first is used and the others are left unchanged. If there is no study column, the first text column that is not a rating or an identifier is used.

Value Standardization (Fuzzy Mapping): Judgment inputs are converted to lowercase, trimmed, and hyphens, underscores, dots and slashes are read as spaces. The mapping logic is as follows:

- **"Definitely Low"/"Definitely Yes"**: "definitely yes", "definitely low", "dy", "dl", "yes", or "y".
- **"Probably Low"/"Probably Yes"**: "probably yes", "probably low", "py", or "pl".
- **"Probably High"/"Probably No"**: "probably no", "probably high", "pn", or "ph".
- **"Definitely High"/"Definitely No"**: "definitely no", "definitely high", "dn", "dh", "no", or "n".
- **"Not applicable"**: "not applicable" or "n/a".
- Empty cells become `NA` (drawn as "Missing" by `eclipseplot`).
- Any other text is kept and reported (drawn as "Missing").

Value

A standardized data frame with the official ROBUST-RCT wording (Step 1: "Definitely Yes" ... "Definitely No"; Step 2 and optional items: "Definitely Low" ... "Definitely High"; or "Not applicable") and column names (pmid, study, itemX_stepY, optX) for use in the eclipse plot. Empty cells are returned as NA; unrecognised values are kept unchanged and reported in a warning, so they can be corrected.

Note

Step 1 columns use the Step 1 wording (Yes/No) and Step 2 and optional columns use the Step 2 wording (Low/High). Both are drawn with the same colours.

Examples

```
data(messy)
messy
organized <- eclipsedf(messy)
organized
```

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

Description

An independent visual representation of risk-of-bias assessments of randomized controlled trials doi:10.31222/osf.io/58yda_v1, based on the ROBUST-RCT framework 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",
  opt_labels = NULL
)
```

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 `FALSE`, displays only the step 2 (judgment).
`optionals`	Logical. If `TRUE`, includes the optional items (columns `opt1`, `opt2`, ...; any number).
`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 (or a single number between 0 and 1). Relative widths (horizontal) or heights (vertical) for plot and legend. By default they are computed from the number of studies.
`opt_prefix`	Character. Text for optional items in the plot. Default is "o".
`opt_labels`	Names of the optional items shown in the legend, for this plot only. Either a vector in the order of the optional items (`c("Selective reporting", "Baseline imbalance")`), or a named vector naming the column or the plot label (`c(opt3 = "Selective reporting")` or `c(o1 = "Selective reporting")`). Items not named keep the default. Default `NULL`: "Optional item 1", "Optional item 2", ...

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 "Not applicable".
- **Optional Items:** If optionals = TRUE, columns named 'optX' (X = 1, 2, 3, ...). Legacy formats ('optX_step1' and 'optX_step2') are supported, but only Step 2 will be plotted. In the plot, the optional items are numbered in sequence (o1, o2, ...) in the order of their column numbers, whatever those numbers are: a file with only 'opt3' and 'opt7' shows o1 and o2. Name them with `opt_labels`.

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".

Values are matched without regard to case, extra spaces or hyphens, and the abbreviations listed in `eclipsedf` are accepted. Empty cells and unrecognised values are drawn as **"Missing"** (white with a dashed outline) and reported in a warning; they are never shown as "Not applicable". Missing core columns are also drawn as "Missing". Studies are placed in the order of the rows, so two studies with the same label are both shown. Use `check_eclipse_data` to list all problems before plotting.

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. The recommended size in inches is stored in attr(x, "eclipse_size") and is used by save_eclipseplot when no size is given.

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)
- Missing / unrecognised: White with a dashed grey outline

Author(s)

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

Yohan Casiraghi (Contributor)

Sofia Simoni Rossi Fermo (Contributor)

Adolfo Moraes de Souza (Contributor)

Maicon Falavigna (Contributor)

Patricia Klarmann Ziegelmann (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,
            opt_labels = c("Selective reporting", "Baseline imbalance", "Other bias"))

# 3b. Naming only one optional item (by column or plot label); the others keep the default
```

```
eclipseplot(sample_long, optionals = TRUE, opt_labels = c(opt2 = "Baseline imbalance"))

# 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")

# 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"). Ratings are standardised with `eclipsedf`.

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.

See Also

`read_eclipse`, which reads both this spreadsheet and plain tables.

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

Description

Get path to example files (.csv)

Usage

```
get_eclipse_data(name = c("brief", "long", "messy"))
```

Arguments

name	Name of the file: "brief", "long" or "messy".

Value

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

Examples

```
get_eclipse_data("brief")
```

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.

read_eclipse *Read an assessment file (.csv, .tsv, .txt, .xlsx or .xls)*

Description

Reads a file in the eclipseplot format or the standard ROBUST spreadsheet. For text files, the encoding (UTF-8 or Windows-1252/Latin-1) and the separator (comma, semicolon or tab) are detected automatically, so CSV files saved by Excel in any language can be used directly. For Excel files, the standard ROBUST spreadsheet is detected and converted with `from_xlsx`; any other sheet is read with its first row as column names.

Usage

```
read_eclipse(path, clean = TRUE, sheet = 1)
```

Arguments

`path`	Path to the file.
`clean`	Logical. If `TRUE` (default), the result is passed to `eclipsedf` to standardise column names and ratings.
`sheet`	Sheet name or index for Excel files (default 1).

Value

A data frame. All columns are read as text.

Examples

```
df <- read_eclipse(get_eclipse_data("brief"))
head(df, 2)
```

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 6 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 three optional items (opt1, opt2, opt3). Note: in this example, Step 2 of items 1 and 2 is often not the equivalent of Step 1; `check_eclipse_data(sample_long)` lists these rows.

Usage

```
sample_long
```

Format

A data frame with 26 rows and 16 variables (pmid, study, the 11 core item columns and opt1 to opt3).

---

save_eclipseplot *Export the eclipseplot to a file*

Description

Export the eclipseplot to a file

Usage

```
save_eclipseplot(plot_object, filename, width = NULL, height = NULL, dpi = 300)
```

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", "plot.pdf", "plot.tiff").
`width`	Width in inches. Default `NULL` uses the size recommended by `eclipseplot()` (it grows with the number of studies), or 14 if unknown.
`height`	Height in inches. Default `NULL` uses the recommended size, or 9.8 if unknown.
`dpi`	Resolution (Default is 300, suitable for journals).

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

∗ **datasets**
messy, 8
sample_brief, 9
sample_long, 10

check_eclipse_data, 2, *5*

eclipsedf, *2*, 3, *5*, *7*, *9*
eclipseplot, 4

from_xlsx, 7, *9*

get_eclipse_data, 8

messy, 8

read_eclipse, *7*, 9

sample_brief, 9
sample_long, 10
save_eclipseplot, 10