Package: miniMeta (via r-universe)

November 20, 2026

Type Package

Title Web Application to Run Meta-Analyses

Version 0.3.3

Date 2026-11-20

Maintainer Theodore Lytras <thlytras@gmail.com>

Imports shiny, shinyjs, shinyWidgets, colourpicker, rhandsontable, metafor, markdown, WriteXLS, readxl, jsonlite, grDevices, methods, stats

Depends meta (>= 7.0-0), R (>= 4.0.0)

Description Shiny web application to run meta-analyses. Essentially a graphical front-end to package 'meta' for R. Can be useful as an educational tool, and for quickly analyzing and sharing meta-analyses. Provides output to quickly fill in GRADE (Grading of Recommendations, Assessment, Development and Evaluations) Summary-of-Findings tables. Importantly, it allows further processing of the results inside R, in case more specific analyses are needed.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/thlytras/miniMeta

BugReports https://github.com/thlytras/miniMeta/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Theodore Lytras [aut, cre] (ORCID: <https://orcid.org/0000-0002-4146-4122>)

Repository https://cran.r-universe.dev

Date/Publication 2026-11-20 13:13:18 UTC

RemoteUrl https://github.com/cran/miniMeta

RemoteRef HEAD

RemoteSha 500aa34dbacb1bda70bd77dcf7d4312c3f3b80a0

Contents

analysisOptions 2
as.source 3
example_analyses 3
forest.miniMeta 4
is.miniMeta 5
is.miniMeta.obs 5
is.miniMeta.rct 6
miniMeta 6
parseArguments 7
plotOptions 8
sampleSizeBin 8
sampleSizeCont 9

Index **11**

`analysisOptions` *Get analytical options from miniMeta object*

Description

This function returns the analytical options stored in a miniMeta object, as a named list of arguments, for further processing.

Usage

```
analysisOptions(x, data = FALSE)
```

Arguments

`x`	An object of class `miniMeta`
`data`	If `TRUE`, the data associated with the meta-analysis are included in the return value. This allows you for example to run: `do.call(metabin, analysisOptions(x, data=TRUE))` or `do.call(metagen, analysisOptions(x, data=TRUE))` (depending on the contents of x), in order to re-run the meta-analysis.

Value

A named list of arguments corresponding to the arguments of `metagen` or `metabin`.

Examples

```
analysisOptions(example_miniMeta_rct)
```

`as.source` *Return miniMeta analysis as source code*

Description

Returns an entire miniMeta analysis in an R source code format. This provides a basis for further processing the results exported from miniMeta, using R code, in order to perform more elaborate or more specific analyses.

Usage

```
as.source(x)
```

Arguments

`x`	An object of class `miniMeta`

Value

A character vector of length one, containing R code that loads the data, runs the meta-analysis, and plots a forest plot. You can save this in a text file using `writeLines`.

Examples

```
fname <- tempfile("my_analysis", fileext = ".R")
fname
# Writes the miniMeta analysis to an R script
writeLines(as.source(example_miniMeta_rct), fname)
```

`example_analyses` *Example miniMeta analyses*

Description

These are example miniMeta meta-analyses, with study data taken from Lytras et al, 2014. Object `example_miniMeta_rct` contains a meta-analysis of Randomized Controlled Trials (RCTs), and `example_miniMeta_obs` a meta-analysis of observational studies.

Usage

```
example_miniMeta_obs

example_miniMeta_rct
```

Format

Objects of class `miniMeta`

An object of class `miniMeta` (inherits from `list`) of length 4.

An object of class `miniMeta` (inherits from `list`) of length 4.

References

Lytras T, Nikolopoulos G, Bonovas S. World J Gastroenterol 2014;20(7):1858-70 (PubMed)

`forest.miniMeta` *Forest plot for miniMeta objects*

Description

Draws a forest plot for a miniMeta object using the options stored in the object

Usage

```
## S3 method for class 'miniMeta'
forest(x, ...)
```

Arguments

`x`	An object of class `miniMeta`
`...`	Further arguments passed to or from other methods

Examples

```
forest(example_miniMeta_obs)
```

is.miniMeta *Is this a miniMeta object?*

Description

This function checks whether this is a valid miniMeta object

Usage

```
is.miniMeta(x)
```

Arguments

x	An object of class `miniMeta`

Value

`TRUE` if it is a valid miniMeta object, `FALSE` if it is not.

Examples

```
is.miniMeta(example_miniMeta_obs) # returns TRUE
is.miniMeta(example_miniMeta_rct) # returns TRUE
```

is.miniMeta.obs *Is this a miniMeta object for observational studies?*

Description

This function checks whether this is a valid miniMeta object holding a a meta-analysis of observational studies.

Usage

```
is.miniMeta.obs(x)
```

Arguments

x	An object of class `miniMeta`

Value

`TRUE` if it is a valid miniMeta object holding a meta-analysis of observational studies, `FALSE` if it is not.

Examples

```
is.miniMeta.obs(example_miniMeta_obs) # returns TRUE
is.miniMeta.obs(example_miniMeta_rct) # returns FALSE
```

`is.miniMeta.rct` *Is this a miniMeta object for RCTs?*

Description

This function checks whether this is a valid miniMeta object holding a a meta-analysis of Randomized Controlled Trials (RCTs).

Usage

```
is.miniMeta.rct(x)
```

Arguments

`x`	An object of class `miniMeta`

Value

`TRUE` if it is a valid miniMeta object holding a meta-analysis of Randomized Controlled Trials (RCTs), `FALSE` if it is not.

Examples

```
is.miniMeta.rct(example_miniMeta_obs) # returns FALSE
is.miniMeta.rct(example_miniMeta_rct) # returns TRUE
```

`miniMeta` *Launch miniMeta in your browser*

Description

This function lanuches miniMeta in your browser

Usage

```
miniMeta()
```

Examples

```
## Not run:
miniMeta()

## End(Not run)
```

`parseArguments` *Parse arguments from a comma-separated list*

Description

Read a comma-separated list of arguments (as a character string), parse them, and return as a named R list. This function is used in miniMeta to parse arguments for forest.meta() when given as a string.

Usage

```
parseArguments(x)
```

Arguments

`x`	A character vector (of length one) containing the arguments. All should be named.

Details

The string is parsed but never evaluated as arbitrary R code: only literal values are accepted (numbers, strings, `TRUE`/`FALSE`, `NA`, `NULL`, `Inf`, `NaN`), optionally combined with `c()` or `list()`, and unary minus/plus on numbers. Anything else (function calls, variables, other operators) is rejected.

Value

A named list of arguments, or an object of class "try-error" on failure.

Examples

```
parseArguments('col.diamond="red", sm="RR", common=FALSE')
```

plotOptions *Get forest plot options from miniMeta object*

Description

This function returns the forest plot options stored in a miniMeta object, as a named list of arguments, for further processing. This allows finer control than directly plotting using the `forest.miniMeta` method. See the example below.

Usage

```
plotOptions(x)
```

Arguments

x	An object of class `miniMeta`

Value

A named list of arguments corresponding to the arguments of `forest.meta`.

Examples

```
## Not run:
# Extract the plot options from the miniMeta object
plot_opts <- plotOptions(example_miniMeta_obs)
# Call directly the forest.meta method, with all plot options
do.call(forest, c(x=list(example_miniMeta_obs$meta), plot_opts))

# Equivalently, call the forest.miniMeta method directly
forest(example_miniMeta_obs)

## End(Not run)
```

sampleSizeBin *Sample size calculator for binary outcomes*

Description

Calculates sample size for a trial with a binomial outcome, for a given power and false positive rate.

Usage

```
sampleSizeBin(cer, RRR = 25, ier = NULL, a = 0.05, b = 0.2, K = 1)
```

Arguments

cer	Control group event rate, a value between 0 and 1. All should be named.
RRR	Relative Risk Reduction (%) in the intervention group.
ier	Intervention group event rate, a value between 0 and 1 If NULL, it is calculated from RRR. If non-NULL, the value of this argument is used and RRR is ignored.
a	False positive rate (alpha). Defaults to 0.05 (5%).
b	False negative rate (beta). Defaults to 0.2. Power is one minus beta; thus the default is 80% power.
K	Ratio of intervention group size to control group size. Defaults to 1, meaning both groups have the same size. Set to infinity (Inf) in order to calculate sample size for a *single-group study*, see details below.

Value

An integer vector of length 2, with the sample sizes for the control and intervention groups.

If K=Inf, then the sample size calculation is not for a study with two groups, but for a single-group study in which a fixed known population event rate is assumed. In that case, argument cer represents the population event rate, and ier the study event rate that it we anticipate. And the return value is a single value, i.e. the sample size of the study.

Examples

```
# Sample size for a trial with 40\% control event rate and 1:1 randomization,
# aiming to show a Relative Risk Reduction of 30\% with 80\% power.
sampleSizeBin(0.4, RRR=30)

# Sample size for a single-group study aiming to show an event rate of 20\%
# against a population event rate of 10\%, with 90\% power.
sampleSizeBin(0.1, ier=0.2, b=0.1, K=Inf)
```

---

sampleSizeCont *Sample size calculator for continuous outcomes*

Description

Calculates sample size for a trial with a continuous outcome, for a given power and false positive rate.

Usage

```
sampleSizeCont(Dm, SD, a = 0.05, b = 0.2, K = 1)
```

Arguments

Dm	Anticipated absolute difference in means between the two groups (intervention and control).
SD	Anticipated standard deviation for the outcome.
a	False positive rate (alpha). Defaults to 0.05 (5%).
b	False negative rate (beta). Defaults to 0.2. Power is one minus beta; thus the default is 80% power.
K	Ratio of intervention group size to control group size. Defaults to 1, meaning both groups have the same size. Set to infinity (Inf) in order to calculate sample size for a *single-group study*, see details below.

Value

An integer vector of length 2, with the sample sizes for the control and intervention groups.

If K=Inf, then the sample size calculation is not for a study with two groups, but for a single-group study in which we try to show a difference from a fixed known population mean. In that case, argument Dm represents the absolute difference between the study mean and population mean, rather than the difference in means between two groups. And the return value is a single value, i.e. the sample size of the study.

Examples

```
# Sample size for a trial with 2:1 randomization, aiming to show a mean
# difference of 2 for a continuous outcome with a standard deviation of 3,
# with 90\% power.
sampleSizeCont(2, 3, b=0.1, K=2)

# Similar for a single-group study aiming to show a difference of 2 against
# a known population mean.
sampleSizeCont(2, 3, b=0.1, K=Inf)
```

Index

∗ **datasets**
example_analyses, 3

analysisOptions, 2
as.source, 3

example_analyses, 3
example_miniMeta_obs
*(*example_analyses*)*, 3
example_miniMeta_rct
*(*example_analyses*)*, 3

forest.meta, *8*
forest.miniMeta, 4, *8*

is.miniMeta, 5
is.miniMeta.obs, 5
is.miniMeta.rct, 6

metabin, *2*
metagen, *2*
miniMeta, 6

parseArguments, 7
plotOptions, 8

sampleSizeBin, 8
sampleSizeCont, 9

writeLines, *3*