Package ‘metagroup’

September 10, 2025

Type Package

Title Meaningful Grouping of Studies in Meta-Analysis

Version 1.0.2

Date 2025-08-31

Author Ahmed Abdelmageed [cre, aut] (ORCID: <https://orcid.org/0009-0002-7902-690X>)

Maintainer Ahmed Abdelmageed <ahmedelsaeedmassad@gmail.com>

URL https://github.com/asmpro7/metagroup/

Description Performs meaningful subgrouping in a meta-analysis. This is a two-step process; first, use the iterative grouping functions (e.g., mgbin(), mgcont()) to partition studies into statistically homogeneous clusters based on their effect size data. Second, use the meaning() function to analyze these new subgroups and understand their composition based on study-level characteristics (e.g., country, setting). This approach helps to uncover hidden structures in meta-analytic data and provide a deeper interpretation of heterogeneity.

License GPL (>= 3)

Encoding UTF-8

Imports dplyr, ggplot2, magrittr, meta, rlang

RoxygenNote 7.3.2

NeedsCompilation no

Repository CRAN

Date/Publication 2025-09-10 08:00:13 UTC

Contents

metagroup-package 2
grouped 2
meaning 3
mgbin 4

mgcont 5
mgcor 6
mggen 7
mginc 8
mgmean 9
mgprop 10
mgrate 11

Index **12**

`metagroup-package` *metagroup: Meaningful Grouping of Studies in Meta-Analysis*

Description

The ‘metagroup‘ package provides a suite of tools to perform meaningful subgrouping in a meta-analysis. This is a two-step process: first, use the iterative grouping functions (e.g., ‘mgbin()‘, ‘mgcont()‘) to partition studies into statistically homogeneous clusters. Second, use the ‘meaning()‘ function to analyze these new subgroups and understand their composition based on study-level characteristics (e.g., country, setting).

Author(s)

Ahmed Abdelmageed `<ahmedelsaeedmassad@gmail.com>`

See Also

Useful functions:

- `mgbin`, `mgcont`, `mgcor` for grouping studies.
- `meaning` for interpreting the new subgroups.

`grouped` *Print and Plot Methods for ’grouped’ Objects*

Description

S3 methods for objects of class ‘grouped‘ returned by the ‘meaning()‘ function.

- ‘print.grouped‘ provides a concise summary view of the results.
- ‘plot.grouped‘ creates a faceted bar chart to visualize the composition of each subgroup.

Usage

```
## S3 method for class 'grouped'
print(x, ...)

## S3 method for class 'grouped'
plot(x, ...)
```

Arguments

`x`	An object of class ‘grouped‘.
`...`	Additional arguments (currently unused).

Value

The ‘print‘ method is called for its side-effect of printing a summary table to the console.

The ‘plot‘ method is called for its side-effect of generating a faceted bar chart in the current graphics device.

Both methods invisibly return the original object ‘x‘.

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

`meaning` *Explore Composition of Homogeneous Study Subgroups*

Description

After grouping studies using a meta-grouping function (e.g., ‘mgbin‘), this function analyzes the composition of each subgroup based on specified categorical variables. It helps to give "meaning" to the statistically derived groups.

Usage

```
meaning(data, subgroup = "subgroup", variables, min = 3)
```

Arguments

`data`	A data frame or the list object returned by a meta-grouping function. If a list, the function will automatically use the ‘data‘ element.
`subgroup`	A string specifying the name of the subgroup column. Defaults to ‘"subgroup"‘.
`variables`	A character vector of column names (categorical variables) to analyze.
`min`	An integer specifying the minimum number of studies a group must have to be included in the final summarized output. Defaults to 3.

Details

For each subgroup and each specified variable, the function calculates the percentage distribution of its categories. It identifies the most frequent (dominant) category and performs a chi-squared test to assess if the distribution of categories within that group is significantly different from random.

Value

An S3 object of class ‘grouped‘, which is a list containing three data frames:

- ‘detailed‘: Contains the percentage distribution for all categories of all variables within every group.
- ‘up.hand‘: A subset of ‘detailed‘, showing only the dominant category for each variable in each group.
- ‘final‘: A summarized version of ‘up.hand‘, showing the single most dominant characteristic for each group that meets the ‘min‘ size threshold. This is the default print output.

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`mgbin`, `mgcont`, `mgcor`

Examples

```
# Create a sample dataset of studies
study_data <- data.frame(
  author = letters[1:10],
  country = sample(c("USA", "China", "Egypt"), 10, replace = TRUE),
  setting = sample(c("Hospital", "Community"), 10, replace = TRUE),
  subgroup = c("group 1", "group 2", "group 1", "group 1", "group 2",
               "group 2", "group 1", "group 3", "group 2", "group 2")
)

# Analyze the composition of the subgroups
meaning_result <- meaning(study_data, variables = c("country", "setting"))

# The default print shows the 'final' summary table
print(meaning_result)

# Access detailed results
# meaning_result$detailed
```

`mgbin` *Group Studies with Binary Outcome Data by Homogeneity*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis with binary outcome data.

Usage

```
mgbin(data, event.e, n.e, event.c, n.c, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`event.e`	A vector of event counts in the experimental group.
`n.e`	A vector of sample sizes in the experimental group.
`event.c`	A vector of event counts in the control group.
`n.c`	A vector of sample sizes in the control group.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metabin‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metabin‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed `<ahmedelsaeedmassad@gmail.com>`

See Also

`meaning`

`mgcont` *Group Studies with Continuous Outcome Data by Homogeneity*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis with continuous outcome data.

Usage

```
mgcont(data, n.e, mean.e, sd.e, n.c, mean.c, sd.c, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`n.e`	A vector of sample sizes in the experimental group.
`mean.e`	A vector of means in the experimental group.
`sd.e`	A vector of standard deviations in the experimental group.
`n.c`	A vector of sample sizes in the control group.
`mean.c`	A vector of means in the control group.
`sd.c`	A vector of standard deviations in the control group.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metacont‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metacont‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

`mgcor` *Group Studies with Correlation Data by Homogeneity*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis with correlation data.

Usage

```
mgcor(data, cor, n, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`cor`	A vector of correlation coefficients.
`n`	A vector of corresponding sample sizes.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metacor‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metacor‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

`mggen` *Group Studies Using the Generic Inverse Variance Method*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final generic inverse variance meta-analysis.

Usage

```
mggen(data, TE, seTE, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`TE`	A vector of treatment effects.
`seTE`	A vector of the standard errors of the treatment effects.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metagen‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metagen‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <ahmedelsaeedmassad@gmail.com>

See Also

`meaning`

`mginc` Group Studies with Incidence Rate Data by Homogeneity

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis of incidence rate data.

Usage

```
mginc(data, event.e, time.e, event.c, time.c, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`event.e`	A vector of event counts in the experimental group.
`time.e`	A vector of person-time at risk in the experimental group.
`event.c`	A vector of event counts in the control group.
`time.c`	A vector of person-time at risk in the control group.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metainc‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metainc‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

`mgmean` *Group Studies with Single Means by Homogeneity*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis of single means.

Usage

```
mgmean(data, mean, sd, n, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`mean`	A vector of means.
`sd`	A vector of standard deviations.
`n`	A vector of sample sizes.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metamean‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metamean‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

`mgprop` *Group Studies with Single Proportions by Homogeneity*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis of single proportions.

Usage

```
mgprop(data, event, n, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`event`	A vector of event counts.
`n`	A vector of corresponding sample sizes.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metaprop‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metaprop‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

`mgrate` *Group Studies with Single Incidence Rates by Homogeneity*

Description

This function iteratively assigns studies to subgroups based on a homogeneity test. The goal is to create statistically homogeneous groups of studies before performing a final meta-analysis of single incidence rates.

Usage

```
mgrate(data, event, time, studlab, ...)
```

Arguments

`data`	A data frame containing the meta-analysis data.
`event`	A vector of event counts.
`time`	A vector of person-time at risk.
`studlab`	A vector of study labels.
`...`	Additional arguments passed on to ‘meta::metarate‘.

Details

The algorithm starts with a single study in "group 1". It then processes each subsequent study, attempting to place it in an existing group. A study is added to a group only if its inclusion does not result in significant within-group heterogeneity. If no suitable group is found, a new one is created.

Value

A list containing the final data with subgroup assignments (‘data‘), the final ‘metarate‘ model (‘model‘), and the number of attempts (‘attempts‘).

Author(s)

Ahmed Abdelmageed <`ahmedelsaeedmassad@gmail.com`>

See Also

`meaning`

Index

∗ **package**
    `metagroup-package`, 2

`grouped`, 2

`meaning`, *2, 3*, 3, *5–11*
`metagroup` *(*`metagroup-package`*)*, 2
`metagroup-package`, 2
`mgbin`, *2*, *4*, 4
`mgcont`, *2*, *4*, 5
`mgcor`, *2*, *4*, 6
`mggen`, 7
`mginc`, 8
`mgmean`, 9
`mgprop`, 10
`mgrate`, 11

`plot.grouped` *(*`grouped`*)*, 2
`print.grouped` *(*`grouped`*)*, 2