

Package: MAIVE (via r-universe)

October 14, 2026

Type Package

Title Meta Analysis Instrumental Variable Estimator

Encoding UTF-8

Version 0.5.0

RoxygenNote 7.3.3

URL <https://meta-analysis.cz/maive/>, <https://github.com/PetrCala/MAIVE>

BugReports <https://github.com/PetrCala/MAIVE/issues>

Description Meta-analysis traditionally assigns more weight to studies with lower standard errors, assuming higher precision. However, in observational research, precision must be estimated and is vulnerable to manipulation, such as p-hacking, to achieve statistical significance. This can lead to spurious precision, invalidating inverse-variance weighting and bias-correction methods like funnel plots. Common methods for addressing publication bias, including selection models, often fail or exacerbate the problem. This package introduces an instrumental variable approach to limit bias caused by spurious precision in meta-analysis. Methods are described in 'Irsova et al.' (2025) [doi:10.1038/s41467-025-63261-0](https://doi.org/10.1038/s41467-025-63261-0).

License MIT + file LICENSE

Depends R (>= 4.0.0)

Imports stats, cli, clubSandwich

Suggests testthat, knitr, rmarkdown, varhandle, metafor, dplyr

VignetteBuilder knitr

LazyData true

Repository <https://petrcala.r-universe.dev>

Date/Publication 2026-09-14 13:20:00 UTC

RemoteUrl <https://github.com/petrcala/maive>

RemoteRef HEAD

RemoteSha 7875325404008a7f94233d7af8ed81fd2f0c467a

Contents

get_funnel_plot	2
maive	3
maive_from_metafor	6
waive	7
Index	10

get_funnel_plot	<i>Draw a funnel plot (base graphics)</i>
-----------------	---

Description

High-level wrapper that accepts the original input data and the result returned by ‘maive()’ / ‘waive()’, prepares the necessary vectors/metadata, and draws the funnel plot on the currently active graphics device.

Usage

```
get_funnel_plot(dat, result, instrument = NULL, model_type = "MAIVE")
```

Arguments

dat	Data frame containing at least numeric columns ‘bs’ (effect sizes) and ‘sebs’ (standard errors).
result	Result list returned by ‘maive()’ or ‘waive()’.
instrument	Optional indicator (0/1). If ‘NULL’, inferred from ‘result’.
model_type	Label used for plot text/legend (e.g., “MAIVE” or “WAIVE”).

Details

Device management (PNG/SVG/PDF, resolution, base64 encoding) is intentionally left to the caller.

Value

Invisibly returns ‘NULL’.

maive

R code for MAIVE

Description

R package for MAIVE: "Spurious Precision in Meta-Analysis of Observational Research" by Zuzana Irsova, Pedro Bom, Tomas Havranek, Petr Cala, and Heiko Rachinger.

Usage

```
maive(
  dat,
  method,
  weight,
  instrument,
  studylevel,
  SE,
  AR,
  first_stage = 1L,
  estimate = NULL,
  se = NULL,
  n = NULL,
  study_id = NULL,
  seed = 123
)
```

Arguments

<code>dat</code>	Data frame with columns <code>bs</code> , <code>sebs</code> , <code>Ns</code> , <code>study_id</code> (optional). Column names can be remapped with <code>estimate</code> , <code>se</code> , <code>n</code> , and <code>study_id</code> .
<code>method</code>	1 FAT-PET, 2 PEESE, 3 PET-PEESE, 4 EK.
<code>weight</code>	0 no weights, 1 standard weights, 2 MAIVE adjusted weights, 3 study weights.
<code>instrument</code>	1 yes, 0 no.
<code>studylevel</code>	Correlation at study level: 0 none, 1 fixed effects, 2 cluster.
<code>SE</code>	SE estimator: 0 CR0 (Huber-White), 1 CR1 (Standard empirical correction), 2 CR2 (Bias-reduced estimator), 3 wild bootstrap.
<code>AR</code>	Anderson Rubin corrected CI for weak instruments (available for unweighted and MAIVE-adjusted weight versions of PET, PEESE, PET-PEESE, not available for fixed effects): 0 no, 1 yes.
<code>first_stage</code>	First-stage specification for the variance model: 1 log (default), 0 levels. The log stage regresses $\log(\text{sebs}^2)$ on $\log(Ns)$ with a smearing retransformation, so the fitted variance is always positive. The levels stage regresses sebs^2 on $1/Ns$ and is the specification in the published paper; pass <code>first_stage = 0</code> to reproduce it.
<code>estimate</code>	Optional column name to use instead of <code>'bs'</code>

se	Optional column name to use instead of 'sebs'
n	Optional column name to use instead of 'Ns'
study_id	Optional column name for study identifiers. When not supplied, a column named study_id is used if present; otherwise, if dat has four or more columns, the fourth column is used as the study identifier and a warning names the column. Columns already mapped through estimate, se, or n are never picked: when the fourth column is one of them, the single remaining column is used instead, and when several remain an error asks for study_id (at studylevel = 0 the fallback is skipped silently, since no identifier is needed). Any fourth column (a moderator, a year) would otherwise drive the study dummies and clustering at every studylevel other than 0, so name the column explicitly or drop it. When studylevel is not 0 and no identifier can be found, a warning says the study-level options have no effect.
seed	Seed for the wild bootstrap when SE = 3. Use NULL to avoid setting a seed (results depend on the current RNG state). Default is 123 for historical reproducibility.

Details

Guided, interactive workflow available at <https://www.easymeta.org>.

Data dat can be imported from an Excel file via: `dat <- read_excel("inputdata.xlsx")` or from a csv file via: `dat <- read.csv("inputdata.csv")` It should contain:

- Estimates: bs
- Standard errors: sebs
- Number of observations: Ns
- Optional: study_id

Default option for MAIVE: MAIVE-PET-PEESE, unweighted, instrumented, cluster SE, wild bootstrap, AR, log first stage.

The levels first stage (`first_stage = 0`) regresses the squared standard errors on $1/N$ by ordinary least squares, so an individual fitted variance can be negative even when the intercept is not. Such an estimate has no instrumented standard error and is excluded from every quantity built on it: the MAIVE-adjusted weights, the second-stage regressions (PET, PEESE, PET-PEESE, EK), the first-stage F-test, the Hausman comparison, the Anderson-Rubin intervals, and the wild bootstrap. The first stage itself is fitted on all rows; every reported statistic then uses the same remaining rows. A warning reports the number of excluded estimates, which is also returned as `n_excluded` with their positions in `excluded_rows`. The log first stage (`first_stage = 1`) cannot fit a negative variance, so it never excludes an estimate.

Value

- beta: MAIVE meta-estimate
- SE: MAIVE standard error
- F-test: heteroskedastic robust F-test of the first step instrumented SEs
- beta_standard: point estimate from the conventional (non-instrumented, inverse-variance weighted) fit of the method chosen

- SE_standard: standard error from the same conventional fit as beta_standard
- Hausman: Hausman type test: comparison between MAIVE and standard version
- Chi2: 5
- SE_instrumented: instrumented standard errors, one per input row; NA for estimates excluded because their fitted variance was not positive
- AR_CI: Anderson-Rubin confidence interval for weak instruments
- pub_bias_p-value: p-value of test for publication bias / p-hacking based on instrumented FAT
- egger_coef: Egger Coefficient (PET estimate)
- egger_se: Egger Standard Error (PET standard error)
- egger_boot_ci: Confidence interval for the Egger coefficient using the selected resampling scheme
- egger_ar_ci: Anderson-Rubin confidence interval for the Egger coefficient (when available)
- is_quadratic_fit: Details on quadratic selection and slope behaviour
- ek_structure: Structure of the fitted EK model when method=4: "kink", "linear", or "intercept" (intercept-only degenerate fit); NA for other methods
- boot_result: Boot result
- slope_coef: Slope coefficient
- petpeeese_selected: Which model (PET or PEESE) was selected when method=3 (NA otherwise)
- peeese_se2_coef: Coefficient on SE² when PEESE is the final model (NA otherwise)
- peeese_se2_se: Standard error of the PEESE SE² coefficient (NA otherwise)
- weights: second-stage row scale, one per input row; NA for excluded estimates. Each row of the second-stage regressions is divided by it, so the least-squares weight of an estimate is $1 / \text{weights}^2$. In `waive()` it is also divided by the square root of the exponential-decay weight, so with `weight = 0`, $1 / \text{weights}^2$ is the decay weight itself
- instrument_strength: "strong", "weak", "very_weak", "unknown", or "not_applicable", from the first-stage F-test
- n_excluded: number of estimates excluded because the first stage fitted a non-positive variance for them (0 with the log first stage)
- excluded_rows: positions of the excluded estimates in the input data (after completely empty rows are dropped); `integer(0)` when none

Examples

```
dat <- data.frame(
  bs = c(0.5, 0.45, 0.55, 0.6),
  sebs = c(0.25, 0.2, 0.22, 0.27),
  Ns = c(50, 80, 65, 90)
)

result <- maive(dat,
  method = 3, weight = 0, instrument = 1,
  studylevel = 0, SE = 0, AR = 0
```

```

)

# first_stage = 0 uses the levels first stage from the published paper
result_levels <- maive(dat,
  method = 3, weight = 0, instrument = 1,
  studylevel = 0, SE = 0, AR = 0, first_stage = 0
)

```

maive_from_metafor	<i>Convert metafor objects into MAIVE input data</i>
--------------------	--

Description

Builds the data frame that `maive()` and `waive()` expect (columns `bs`, `sebs`, `Ns`, and optionally `study_id`) from a **metafor** `escalc` data frame or an `rma.uni` model fit.

Usage

```
maive_from_metafor(x, ni = NULL, study_id = NULL)
```

Arguments

<code>x</code>	An <code>escalc</code> data frame (or any data frame with <code>yi</code> and <code>vi</code> columns) or an <code>rma.uni</code> fit.
<code>ni</code>	Optional sample sizes: a numeric vector, or the name of a column in <code>x</code> (or in the data stored with the fit). Overrides any sample sizes carried by <code>x</code> .
<code>study_id</code>	Optional study identifiers: a vector, or the name of a column in <code>x</code> (or in the data stored with the fit). Vectors may have either the length of the original data or the number of rows the fit kept.

Details

The conversion takes the standard error as `sqrt(vi)`, so `metafor`'s sampling variance cannot end up in the `sebs` slot by mistake. Sample sizes are never inferred from the variances; doing so would reintroduce the spurious precision that MAIVE corrects for. They are resolved, in order, from the `ni` argument, an `ni` column (or the fit's `ni`), the `ni` attribute that `escalc()` stamps on the effect size column, and finally the exact two-group total $n1i + n2i$. The column and the attribute are only used when they agree with $n1i + n2i$ wherever both are present: the attribute does not follow a reorder such as `dplyr::arrange()` and `rma()` copies it into the fit, so a stale vector of the right length falls through to the two-group total instead of binding every sample size to the wrong estimate. When none of these is available the function stops and names what it needs.

For `rma.uni` fits the effect sizes, variances, sample sizes, and any vector supplied through `ni` or `study_id` are taken through the same subset and missing-value masks the fit applied, so the rows stay aligned. Multivariate (`rma.mv`), GLMM (`rma.glmm`), and other `rma` subclasses are refused rather than silently flattened, and so are `metafor::trimfill()` fits: their imputed studies would enter MAIVE as observed estimates, so pass the original `rma()` fit.

Value

A data frame with columns `bs`, `sebs`, `Ns`, and, when identifiers were supplied, `study_id`, ready for `maive()` or `waive()`.

Examples

```
dat <- metafor::escalc(
  measure = "SMD",
  m1i = c(2.1, 2.5, 1.9, 2.4, 2.2, 2.6),
  sd1i = c(1.0, 1.2, 0.9, 1.1, 1.0, 1.3),
  n1i = c(40, 50, 30, 45, 60, 35),
  m2i = c(1.8, 2.0, 1.7, 2.1, 1.9, 2.2),
  sd2i = c(1.0, 1.1, 1.0, 1.0, 1.1, 1.2),
  n2i = c(42, 55, 33, 48, 58, 37)
)
maive_dat <- maive_from_metafor(dat)
head(maive_dat)

fit <- metafor::rma(yi, vi, data = dat)
identical(maive_from_metafor(fit), maive_dat)

result <- maive(maive_dat,
  method = 3, weight = 0, instrument = 1,
  studylevel = 0, SE = 0, AR = 0
)
```

waive

WAIVE: More Aggressive Correction for P-Hacking and Spurious Precision

Description

WAIVE (Weighted Adjusted Instrumental Variable Estimator) provides a more aggressive correction for p-hacking and spurious precision by extending MAIVE with exponential-decay weights that downweight both spuriously precise estimates and extreme outliers.

Usage

```
waive(
  dat,
  method,
  weight,
  instrument,
  studylevel,
  SE,
  AR,
  first_stage = 1L,
```

```

estimate = NULL,
se = NULL,
n = NULL,
study_id = NULL,
seed = 123
)

```

Arguments

<code>dat</code>	Data frame with columns <code>bs</code> , <code>sebs</code> , <code>Ns</code> , <code>study_id</code> (optional). Column names can be remapped with <code>estimate</code> , <code>se</code> , <code>n</code> , and <code>study_id</code> .
<code>method</code>	1 FAT-PET, 2 PEESE, 3 PET-PEESE, 4 EK.
<code>weight</code>	0 no weights, 1 standard weights, 2 MAIVE adjusted weights, 3 study weights.
<code>instrument</code>	1 yes, 0 no.
<code>studylevel</code>	Correlation at study level: 0 none, 1 fixed effects, 2 cluster.
<code>SE</code>	SE estimator: 0 CR0 (Huber-White), 1 CR1 (Standard empirical correction), 2 CR2 (Bias-reduced estimator), 3 wild bootstrap.
<code>AR</code>	Anderson Rubin corrected CI for weak instruments (available for unweighted and MAIVE-adjusted weight versions of PET, PEESE, PET-PEESE, not available for fixed effects): 0 no, 1 yes.
<code>first_stage</code>	First-stage specification for the variance model: 1 log (default), 0 levels. The log stage regresses $\log(\text{sebs}^2)$ on $\log(Ns)$ with a smearing retransformation, so the fitted variance is always positive. The levels stage regresses sebs^2 on $1/Ns$ and is the specification in the published paper; pass <code>first_stage = 0</code> to reproduce it.
<code>estimate</code>	Optional column name to use instead of 'bs'
<code>se</code>	Optional column name to use instead of 'sebs'
<code>n</code>	Optional column name to use instead of 'Ns'
<code>study_id</code>	Optional column name for study identifiers. When not supplied, a column named <code>study_id</code> is used if present; otherwise, if <code>dat</code> has four or more columns, the fourth column is used as the study identifier and a warning names the column. Columns already mapped through <code>estimate</code> , <code>se</code> , or <code>n</code> are never picked: when the fourth column is one of them, the single remaining column is used instead, and when several remain an error asks for <code>study_id</code> (at <code>studylevel = 0</code> the fallback is skipped silently, since no identifier is needed). Any fourth column (a moderator, a year) would otherwise drive the study dummies and clustering at every <code>studylevel</code> other than 0, so name the column explicitly or drop it. When <code>studylevel</code> is not 0 and no identifier can be found, a warning says the study-level options have no effect.
<code>seed</code>	Seed for the wild bootstrap when <code>SE = 3</code> . Use <code>NULL</code> to avoid setting a seed (results depend on the current RNG state). Default is 123 for historical reproducibility.

Details

Guided, interactive workflow available at <https://www.easymeta.org>.

For technical details on WAIVE methodology, see: https://meta-analysis.cz/waive_ottawa.pdf

WAIVE combines variance instrumentation (as in MAIVE) with robust downweighting based on first-stage residuals. Studies with negative residuals (spurious precision) or extreme residuals (outliers) receive exponentially reduced influence in the meta-analytic estimate. This makes WAIVE more aggressive than standard MAIVE at correcting for p-hacking and handling outliers.

Value

List with the same structure as `maive()`. See `?maive` for details.

Examples

```
dat <- data.frame(
  bs = c(0.5, 0.45, 0.55, 0.6),
  sebs = c(0.25, 0.2, 0.22, 0.27),
  Ns = c(50, 80, 65, 90)
)

result <- waive(dat,
  method = 3, weight = 0, instrument = 1,
  studylevel = 0, SE = 0, AR = 0
)
```

Index

`get_funnel_plot`, [2](#)

`maive`, [3](#), [6](#), [7](#)

`maive_from_metafor`, [6](#)

`waive`, [6](#), [7](#), [7](#)