meta set — Declare meta-analysis data using generic effect sizes

Description

meta set declares the data in memory to be meta data, informing Stata of key variables and their roles in a meta-analysis. It is used with generic (precomputed) effect sizes specified in the metric closest to normality; see [META] meta esize if you need to compute and declare effect sizes. You must use meta set or meta esize to perform meta-analysis using the meta command; see [META] meta data.

If you need to update some of the meta settings after the data declaration, see [META] meta update. To display current meta settings, use meta query; see [META] meta update.

Quick start

Declare generic effect sizes and their standard errors from individual studies stored in variables es and se

```
meta set es se
```

As above, but request a random-effects meta-analysis where between-study heterogeneity is estimated using the DerSimonian–Laird method instead of the default REML method

```
meta set es se, random(dlaird)
```

Specify a common-effect meta-analysis, study labels stored in a string variable studylab, and label effect sizes as \( \log(\text{HR}) \) in the output

```
meta set es se, common studylab(studylab) eslabel("\log(\text{HR})")
```

Use 90% confidence level, and suppress the display of meta settings for all subsequent meta-analysis commands

```
meta set es se, level(90) nometashow
```

Specify study sizes stored in variable ssize

```
meta set es se, studysize(ssize)
```

Declare generic effect sizes, and compute their standard errors based on the specified 90% CI variables, cil and ciu

```
meta set es cil ciu, civarlevel(90)
```

Menu

Statistics > Meta-analysis
Syntax

Specify generic effect sizes and their standard errors

```
meta set esvar sevar [if] [in] [, options]
```

Specify generic effect sizes and their confidence intervals

```
meta set esvar cilvar ciuvar [if] [in] [, civarlevel(#) civartolerance(#) options]
```

esvar specifies a variable containing the effect sizes, sevar specifies a variable containing standard errors of the effect sizes, and cilvar and ciuvar specify variables containing the respective lower and upper bounds of (symmetric) confidence intervals for the effect sizes. esvar and the other variables must correspond to effect sizes specified in the metric closest to normality, such as log odds-ratios instead of odds ratios.

```
options  Description

Model
random[ (remethod)]  random-effects meta-analysis; default is random(reml)
common  common-effect meta-analysis; implies inverse-variance method
fixed  fixed-effects meta-analysis; implies inverse-variance method

Options
studylab(varname)  variable to be used to label studies in all meta-analysis output
studysize(varname)  total sample size per study
eslabel(string)  effect-size label to be used in all meta-analysis output; default is eslabel(Effect Size)
level(#)  confidence level for all subsequent meta-analysis commands
[no]metashow  display or suppress meta settings with other meta commands

remethod  Description
rem1  restricted maximum likelihood; the default
mle  maximum likelihood
ebayes  empirical Bayes
dlaird  DerSimonian–Laird
sjonkman  Sidik–Jonkman
hedges  Hedges
hschmidt  Hunter–Schmidt
```
Options

civarlevel(#) is relevant only when you specify CI variables cilvar and ciuvar with meta set. It specifies the confidence level corresponding to these variables. The default is civarlevel(95). This option affects the computation of the effect-size standard errors stored in the system variable _meta_se.

Do not confuse civarlevel() with level(). The former affects the confidence level only for the specified CI variables. The latter specifies the confidence level for the meta-analysis.

civartolerance(#) is relevant only when you specify CI variables cilvar and ciuvar with meta set. cilvar and ciuvar must define a symmetric CI based on the normal distribution. civartolerance() specifies the tolerance to check whether the CI is symmetric. The default is civartolerance(1e-6). Symmetry is declared when reldif(ciuvar – esvar,esvar – cilvar) < #.

meta set expects the effect sizes and CIs to be specified in the metric closest to normality, which implies symmetric CIs. Effect sizes and their CIs are often reported in the original metric and with limited precision that, after the normalizing transformation, may lead to asymmetric CIs. In that case, the default of 1e–6 may be too stringent. You may use civartolerance() to loosen the default.

Options random(), common, and fixed declare the meta-analysis model globally throughout the entire meta-analysis; see Declaring a meta-analysis model in [META] meta data. In other words, once you set your meta-analysis model using meta set, all subsequent meta commands will assume that same model. You can update the declared model by using meta update or change it temporarily by specifying the corresponding option with the meta commands. Options random(), common, and fixed may not be combined. If these options are omitted, random(reml) is assumed; see Default meta-analysis model and method in [META] meta data. Also see Meta-analysis models in [META] Intro.

random and random(remethod) specify that a random-effects model be assumed for meta-analysis; see Random-effects model in [META] Intro.

remethod specifies the type of estimator for the between-study variance \( \tau^2 \). remethod is one of reml, mle, ebayes, d Laird, sjonkman, hedges, or hechmidt. random is a synonym for random(reml). Below, we provide a short description for each method based on Veroniki et al. (2016). Also see Declaring a meta-analysis estimation method in [META] meta data.

reml, the default, specifies that the REML method (Raudenbush 2009) be used to estimate \( \tau^2 \). This method produces an unbiased, nonnegative estimate of the between-study variance and is commonly used in practice. Method reml requires iteration.

mle specifies that the ML method (Hardy and Thompson 1996) be used to estimate \( \tau^2 \). It produces a nonnegative estimate of the between-study variance. With a few studies or small studies, this method may produce biased estimates. With many studies, the ML method is more efficient than the REML method. Method mle requires iteration.

ebayes specifies that the empirical Bayes estimator (Berkey et al. 1995), also known as the Paule–Mandel estimator (Paule and Mandel 1982), be used to estimate \( \tau^2 \). From simulations, this method, in general, tends to be less biased than other random-effects methods, but it is also less efficient than reml or d Laird. Method ebayes produces a nonnegative estimate of \( \tau^2 \) and requires iteration.
meta set — Declare meta-analysis data using generic effect sizes

dlaird specifies that the DerSimonian–Laird method (DerSimonian and Laird 1986) be used to estimate $\tau^2$. This method, historically, is one of the most popular estimation methods because it does not make any assumptions about the distribution of random effects and does not require iteration. But it may underestimate the true between-study variance, especially when the variability is large and the number of studies is small. This method may produce a negative value of $\tau^2$ and is thus truncated at zero in that case.

sjonkman specifies that the Sidik–Jonkman method (Sidik and Jonkman 2005) be used to estimate $\tau^2$. This method always produces a nonnegative estimate of the between-study variance and thus does not need truncating at 0, unlike the other noniterative methods. Method sjonkman does not require iteration.

hedges specifies that the Hedges method (Hedges 1983) be used to estimate $\tau^2$. When the sampling variances of effect-size estimates can be estimated without bias, this estimator is exactly unbiased (before truncation), but it is not widely used in practice (Veroniki et al. 2016). Method hedges does not require iteration.

hschmidt specifies that the Hunter–Schmidt method (Schmidt and Hunter 2015) be used to estimate $\tau^2$. Although this estimator achieves a lower MSE than other methods, except ML, it is known to be negatively biased. Method hschmidt does not require iteration.

common specifies that a common-effect model be assumed for meta-analysis; see Common-effect ("fixed-effect") model in [META] Intro. It uses the inverse-variance estimation method; see Meta-analysis estimation methods in [META] Intro. Also see the discussion in [META] meta data about common-effect versus fixed-effects models.

fixed specifies that a fixed-effects model be assumed for meta-analysis; see Fixed-effects model in [META] Intro. It uses the inverse-variance estimation method; see Meta-analysis estimation methods in [META] Intro. Also see the discussion in [META] meta data about fixed-effects versus common-effect models.

studylabel(varname) specifies a string variable containing labels for the individual studies to be used in all applicable meta-analysis output. The default study labels are Study 1, Study 2, ..., Study $K$, where $K$ is the total number of studies in the meta-analysis.

studysize(varname) specifies the variable that contains the total sample size for each study. This option is useful for subsequent meta commands that use this information in computations such as meta funnelplot using the sample-size metric.

eslabel(string) specifies that string be used as the effect-size label in all relevant meta-analysis output. The default label is Effect Size.

level(#) specifies the confidence level, as a percentage, for confidence intervals. It will be used by all subsequent meta-analysis commands when computing confidence intervals. The default is level(95) or as set by set level; see [R] level. After the declaration, you can specify level() with meta update to update the confidence level to be used throughout the rest of the meta-analysis session. You can also specify level() directly with the meta commands to modify the confidence level, temporarily, during the execution of the command.

metashow and nometashow display or suppress the meta setting information in the output of other meta commands. By default, this information is displayed at the top of their output. You can also specify nometashow with meta update to suppress the meta setting output for the entire meta-analysis session after the declaration.
Remarks and examples

Remarks are presented under the following headings:

Overview
Using meta set

Overview

When you perform meta-analysis, it is common for studies included in the meta-analysis to contain precalculated effect sizes, which we refer to as generic effect sizes, such as mean differences, odds ratios, correlations, and hazard ratios. You can use meta set to declare the generic effect sizes specified in the metric closest to normality. (If you have summary data from which effect sizes can be computed, use [META] meta esize instead.)

In addition to effect sizes, their standard errors must be available for meta-analysis. Sometimes, the standard errors are not available, but the confidence intervals (CIs) are. In that case, the standard errors can be computed from the effect-size estimates and CIs. meta set supports both cases. You can supply the variables containing effect sizes and their standard errors, or, instead of the standard errors, you can specify the variables containing the CIs.

When you specify the CI variables, you can specify their corresponding confidence level in the civarlevel() option. (Do not confuse this option with the level() option. The former corresponds to the specified CI variables, whereas the latter specifies the confidence level for the entire meta-analysis.)

Meta-analysis uses effect sizes in a metric that makes them approximately normally distributed such as log odds-ratios instead of odds ratios and log hazard-ratios instead of hazard ratios. As such, meta set expects the effect sizes and measures of their precision to be specified in the metric closest to normality. So, the corresponding standard errors or CIs should be provided in the same metric as effect sizes. For example, if you are working with hazard ratios, you should specify log hazard-ratios with meta set and provide CIs for the log hazard-ratios and not the hazard ratios.

See [META] meta data for more details.
Using meta set

Consider the following fictional meta-analysis dataset:

```
. use https://www.stata-press.com/data/r16/metaset
(Generic effect sizes; fictional data)
```

```
Contains data from https://www.stata-press.com/data/r16/metaset.dta
```

```markdown
| obs: 10 | Generic effect sizes; fictional data |
| vars: 9 |
| 19 Apr 2019 01:28 |
| study | byte | %9.0g | Study ID |
| es | double | %10.0g | Effect sizes |
| se | double | %10.0g | Std. Err. for effect sizes |
| cil | double | %10.0g | 95% lower CI limit |
| ciu | double | %10.0g | 95% upper CI limit |
| ciu90 | double | %10.0g | 90% lower CI limit |
| ciu90 | double | %10.0g | 90% upper CI limit |
| studylab | str23 | %23s | Study label |
| ssize | byte | %9.0g | Study size |
```

Sorted by:

We will use it to describe various usages of the `meta set` command. For examples of declarations of real datasets, see [META] meta data. We assume that `es` contains the effect sizes that are approximately normal (perhaps after a suitable transformation) and that `se`, `cil`, and `ciu` contain their corresponding standard errors and CIs.

Example 1: Declaring effect sizes and standard errors

Meta-analysis datasets often contain precomputed effect sizes and their standard errors. To declare them for meta-analysis using the `meta` commands, we specify the corresponding variables with `meta set`. 
Briefly, *meta set* reports that there are 10 studies, that *es* and *se* are the variables used to declare effect sizes and their standard errors, that the default confidence level is 95%, and more. See *Meta settings with meta set* in [*META*] *meta data* for a detailed description of all settings for this dataset.

We can now use, for example, *meta summarize* to compute the overall effect size (labeled as theta in the output below).

```
. meta summarize
    Effect-size label: Effect Size
    Effect size: es
    Std. Err.: se

Meta-analysis summary
Number of studies = 10
Random-effects model
Heterogeneity:
    tau2 = 0.0157
    I2 (%) = 5.30
    H2 = 1.06

<table>
<thead>
<tr>
<th>Study</th>
<th>Effect Size</th>
<th>[95% Conf. Interval]</th>
<th>% Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study 1</td>
<td>1.480</td>
<td>-0.352</td>
<td>3.311</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.999</td>
<td>-0.933</td>
<td>2.931</td>
</tr>
<tr>
<td>Study 3</td>
<td>1.272</td>
<td>0.427</td>
<td>2.117</td>
</tr>
<tr>
<td>Study 4</td>
<td>1.001</td>
<td>0.750</td>
<td>1.252</td>
</tr>
<tr>
<td>Study 5</td>
<td>1.179</td>
<td>-0.527</td>
<td>2.884</td>
</tr>
<tr>
<td>Study 6</td>
<td>1.939</td>
<td>0.427</td>
<td>3.452</td>
</tr>
<tr>
<td>Study 7</td>
<td>2.377</td>
<td>1.005</td>
<td>3.750</td>
</tr>
<tr>
<td>Study 8</td>
<td>0.694</td>
<td>-0.569</td>
<td>1.956</td>
</tr>
<tr>
<td>Study 9</td>
<td>1.099</td>
<td>-0.147</td>
<td>2.345</td>
</tr>
<tr>
<td>Study 10</td>
<td>1.805</td>
<td>-0.151</td>
<td>3.761</td>
</tr>
<tr>
<td>theta</td>
<td>1.138</td>
<td>0.857</td>
<td>1.418</td>
</tr>
</tbody>
</table>
```

Test of theta = 0: z = 7.95  Prob > |z| = 0.0000
Test of homogeneity: Q = chi2(9) = 6.34  Prob > Q = 0.7054

See [*META*] *meta summarize* for details about this command.
Example 2: Declaring effect sizes and confidence intervals

Continuing with example 1, we find that some meta-analysis datasets contain confidence intervals associated with effect sizes instead of standard errors. In that case, you can specify confidence intervals with `meta set` instead of the standard errors. For example, variables `cil` and `ciu` contain the 95% lower and upper CI limits for the effect sizes stored in variable `es`. We can declare them as follows.

```
.meta set es cil ciu
```

Meta-analysis setting information

Study information
- No. of studies: 10
- Study label: Generic
- Study size: N/A

Effect size
- Type: Generic
- Label: Effect Size
- Variable: es

Precision
- Std. Err.: `_meta_se`
- CI: `[_meta_cil, _meta_ciu]`
- CI level: 95%, controlled by `level()`
- User CI: `[cil, ciu]`
- User CI level: 95%, controlled by `civarlevel()`

Model and method
- Model: Random-effects
- Method: REML

Compared with Std. Err.: in example 1, Std. Err.: under Precision now contains the system variable `_meta_se`; see *System variables* in [META] *meta data*. The standard errors are computed from `cil` and `ciu` and stored in this system variable. The CI values are stored in the corresponding system variables `_meta_cil` and `_meta_ciu`.

The output additionally reports the user-specified CI variables, `cil` and `ciu`, under User CI: and their corresponding confidence level, 95%, under User CI level:. As we will see later, User CI level, controlled by the `civarlevel()` option, and CI level, controlled by the `level()` option, may be different.
Let's now check that we obtain the same results as before using the equivalent CI declaration.

```
.meta summarize

<table>
<thead>
<tr>
<th>Effect Size</th>
<th>[95% Conf. Interval]</th>
<th>% Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study 1</td>
<td>1.480</td>
<td>-0.352</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.999</td>
<td>-0.933</td>
</tr>
<tr>
<td>Study 3</td>
<td>1.272</td>
<td>0.427</td>
</tr>
<tr>
<td>Study 4</td>
<td>1.001</td>
<td>0.750</td>
</tr>
<tr>
<td>Study 5</td>
<td>1.179</td>
<td>-0.527</td>
</tr>
<tr>
<td>Study 6</td>
<td>1.939</td>
<td>0.427</td>
</tr>
<tr>
<td>Study 7</td>
<td>2.377</td>
<td>1.005</td>
</tr>
<tr>
<td>Study 8</td>
<td>0.694</td>
<td>-0.569</td>
</tr>
<tr>
<td>Study 9</td>
<td>1.099</td>
<td>-0.147</td>
</tr>
<tr>
<td>Study 10</td>
<td>1.805</td>
<td>-0.151</td>
</tr>
<tr>
<td>theta</td>
<td>1.138</td>
<td>0.857</td>
</tr>
</tbody>
</table>
```

Test of theta = 0: z = 7.95  Prob > |z| = 0.0000
Test of homogeneity: Q = chi2(9) = 6.34  Prob > Q = 0.7054

In the earlier `meta set`, we assumed that the `cil` and `ciu` variables correspond to the 95% CIs. Although typical, this may not always be the case. You can use the `civarlevel()` option to specify the confidence level of the CI variables. We have variables `cil90` and `ciu90` in our dataset, which contain the 90% CIs for `es`. We can use them in the declaration as long as we also specify the `civarlevel(90)` option.

```
.meta set es cil90 ciu90, civarlevel(90)
```

The `User CI level` now contains 90%. Do not confuse the `civarlevel()` option, whose value is reported in `User CI level`, with the `level()` option, whose value is reported in `CI level`. The former specifies the confidence level corresponding to the declared CI variables. The latter specifies the...
confidence level that will be used to compute various confidence intervals during your meta-analysis session. Note that the system CI variables, _meta_cil and _meta_ciu, always correspond to the confidence level controlled by level().

/meta summarize

Effect-size label: Effect Size
Effect size: es
Std. Err.: _meta_se

Meta-analysis summary

Number of studies = 10
Random-effects model
Method: REML
tau2 = 0.0157
I2 (%) = 5.30
H2 = 1.06

<table>
<thead>
<tr>
<th>Study</th>
<th>Effect Size</th>
<th>[95% Conf. Interval]</th>
<th>% Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study 1</td>
<td>1.480</td>
<td>-0.352</td>
<td>3.311</td>
</tr>
<tr>
<td>Study 2</td>
<td>0.999</td>
<td>-0.933</td>
<td>2.931</td>
</tr>
<tr>
<td>Study 3</td>
<td>1.272</td>
<td>0.427</td>
<td>2.117</td>
</tr>
<tr>
<td>Study 4</td>
<td>1.001</td>
<td>0.750</td>
<td>1.252</td>
</tr>
<tr>
<td>Study 5</td>
<td>1.179</td>
<td>-0.527</td>
<td>2.884</td>
</tr>
<tr>
<td>Study 6</td>
<td>1.939</td>
<td>0.427</td>
<td>3.452</td>
</tr>
<tr>
<td>Study 7</td>
<td>2.377</td>
<td>1.005</td>
<td>3.750</td>
</tr>
<tr>
<td>Study 8</td>
<td>0.694</td>
<td>-0.569</td>
<td>1.956</td>
</tr>
<tr>
<td>Study 9</td>
<td>1.099</td>
<td>-0.147</td>
<td>2.345</td>
</tr>
<tr>
<td>Study 10</td>
<td>1.805</td>
<td>-0.151</td>
<td>3.761</td>
</tr>
</tbody>
</table>

| theta | 1.138 | 0.857 | 1.418 |

Test of theta = 0: z = 7.95
Prob > |z| = 0.0000
Test of homogeneity: Q = chi2(9) = 6.34
Prob > Q = 0.7054

Although the specified CI variables corresponded to the 90% confidence level, the CIs reported by meta summarize are the 95% CIs because the default confidence level is 95%, level(95).

Technical note

As we mentioned earlier, meta set expects the effect sizes and measures of their precision such as CIs to be specified in the metric closest to normality, which implies symmetric CIs. When you specify CIs with meta set, the command checks that the CIs are symmetric within a certain tolerance. The default tolerance is 1e–6.

In practice, effect sizes and their CIs are often reported in the original metric and with limited precision that, after the normalizing transformation, may lead to asymmetric CIs. In that case, the default of 1e–6 may be too stringent. You may loosen the tolerance by specifying the civartolerance() option.
Example 3: Declaring meta-analysis methods and models

In *Declaring a meta-analysis model* in [META] meta data, we describe the importance of choosing the appropriate meta-analysis model and method for the analysis. Here we demonstrate how to specify different meta-analysis models and methods.

From example 1 and as described in *Default meta-analysis model and method* in [META] meta data, the default meta-analysis model and estimation method are random-effects and REML. We can specify a different random-effects method in the random() option. For example, let’s use the DerSimonian–Laird estimation method.

```
.meta set es se, random(dlaird)
```

Meta-analysis setting information

Study information

| No. of studies: 10 |
| Study label: Generic |
| Study size: N/A |

Effect size

| Type: Generic |
| Label: Effect Size |
| Variable: es |

Precision

| Std. Err.: se |
| CI: [_meta_cil, _meta_ciu] |
| CI level: 95% |

Model and method

| Model: Random-effects |
| Method: DerSimonian-Laird |

meta set reports in Method: that the current method is now DerSimonian–Laird.

We can also choose a different meta-analysis model. For example, we can specify a fixed-effects model by using the fixed option.

```
.meta set es se, fixed
```

Meta-analysis setting information

Study information

| No. of studies: 10 |
| Study label: Generic |
| Study size: N/A |

Effect size

| Type: Generic |
| Label: Effect Size |
| Variable: es |

Precision

| Std. Err.: se |
| CI: [_meta_cil, _meta_ciu] |
| CI level: 95% |

Model and method

| Model: Fixed-effects |
| Method: Inverse-variance |

The inverse-variance estimation method is assumed for the fixed-effects model.
We can also specify a common-effect model, although the literature does not recommend starting your meta-analysis with this model.

```
.meta set es se, common
```

Meta-analysis setting information

Study information
No. of studies: 10
Study label: Generic
Study size: N/A

Effect size
Type: Generic
Label: Effect Size
Variable: es

Precision
Std. Err.: se
CI: [_meta_cil, _meta_ciu]
CI level: 95%

Model and method
Model: Common-effect
Method: Inverse-variance

The inverse-variance estimation method is assumed for the common-effect model.

As we describe in *Declaring a meta-analysis model* in [META] *meta data*, some of the meta-analysis will not be available for common-effect models. For example, because a common-effect model implies no heterogeneity, you will not be able to perform tests of small-study effects using `meta bias` in the presence of moderators.

```
.meta bias x, egger
.meta bias with moderators not supported with a common-effect model
```

The declared model is a common-effect model, which assumes no heterogeneity. Specifying moderators that account for potential heterogeneity is not valid in this case. You may override this assumption by specifying one of options `fixed` or `random(remethod)`.

```
r(498);
```

See [META] *meta bias*.

Example 4: Specifying study and effect-size labels, confidence level, and more

In *Declaring display settings for meta-analysis* of [META] *meta data*, we describe the options to control the display from the `meta` commands. Below, we use `studylabel()` and `eslabel()` to specify our own study and effect-size labels, `level(90)` to report the 90% CIs, and `nometashow` to suppress the information about the effect-size and standard-error variables in the output of all `meta` commands.
. meta set es se, studylabel(studylab) eslabel("Mean Diff.") level(90)
> nometashow

Meta-analysis setting information

Study information
- No. of studies: 10
- Study label: studylab
- Study size: N/A

Effect size
- Type: Generic
- Label: Mean Diff.
- Variable: es

Precision
- Std. Err.: se
- CI: [_meta_cil, _meta_ciu]
- CI level: 90%

Model and method
- Model: Random-effects
- Method: REML

If we now run `meta summarize`, we will see the new labels for the studies in the Study column, the effect-size column labeled as Mean Diff., the 90% CIs, and no meta setting information above the table header.

```
. meta summarize

Number of studies = 10
Random-effects model
Method: REML
tau2 = 0.0157
I2 (%) = 5.30
H2 = 1.06

<table>
<thead>
<tr>
<th>Study</th>
<th>Mean Diff.</th>
<th>[90% Conf. Interval]</th>
<th>% Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith et al. (1984)</td>
<td>1.480</td>
<td>-0.057</td>
<td>3.016</td>
</tr>
<tr>
<td>Jones and Miller (1989)</td>
<td>0.999</td>
<td>-0.622</td>
<td>2.620</td>
</tr>
<tr>
<td>Johnson et al. (1991)</td>
<td>1.272</td>
<td>0.563</td>
<td>1.981</td>
</tr>
<tr>
<td>Brown et al. (1995)</td>
<td>1.001</td>
<td>0.790</td>
<td>1.211</td>
</tr>
<tr>
<td>Clark and Thomas (1998)</td>
<td>1.179</td>
<td>-0.252</td>
<td>2.610</td>
</tr>
<tr>
<td>Williams et al. (2003)</td>
<td>1.939</td>
<td>0.670</td>
<td>3.209</td>
</tr>
<tr>
<td>Davis and Wilson (2010)</td>
<td>2.377</td>
<td>1.226</td>
<td>3.529</td>
</tr>
<tr>
<td>Moore and Parker (2014)</td>
<td>0.694</td>
<td>-0.366</td>
<td>1.753</td>
</tr>
<tr>
<td>Miller et al. (2018)</td>
<td>1.099</td>
<td>0.053</td>
<td>2.144</td>
</tr>
<tr>
<td>Assaad et al. (2019)</td>
<td>1.805</td>
<td>0.164</td>
<td>3.446</td>
</tr>
</tbody>
</table>

theta |
1.138 0.902 1.373

theta test = 0: z = 7.95
Prob > |z| = 0.0000

Test of homogeneity: Q = chi2(9) = 6.34
Prob > Q = 0.7054
```
Example 5: Specifying study size

Some analysis such as a funnel plot with sample-size metrics (see [META] meta funnelplot) requires that you specify the sample size for each study with meta set. You can use the studysize() option for this.

```
.meta set es se, studysize(ssize)
```

Meta-analysis setting information

Study information
No. of studies: 10
Study label: Generic
Study size: ssize

Effect size
Type: Generic
Label: Effect Size
Variable: es

Precision
Std. Err.: se
CI: [_meta_cil, _meta_ciu]
CI level: 95%

Model and method
Model: Random-effects
Method: REML

The name of the study-size variable, ssize, is now reported in Study size:

Stored results

meta set stores the following characteristics and system variables:

Characteristics
- `_dtameta_marker`
- `_dtameta_K`
- `_dtameta_studylabel`
- `_dtameta_sudysize`
- `_dtameta_estype`
- `_dtameta_eslabelopt`
- `_dtameta_eslabel`
- `_dtameta_eslabeldb`
- `_dtameta_esvar`
- `_dtameta_esvardb`
- `_dtameta_civarlevel`
- `_dtameta_modellabel`
- `_dtameta_model`
- `_dtameta_methodlabel`
- `_dtameta_method`
- `_dtameta_randomopt`
- `_dtameta_show`
- `_dtameta_datatype`
- `_dtameta_datavars`
meta set — Declare meta-analysis data using generic effect sizes 15

_meta_setcmdline_  
_meta_ifexp_  
_meta_inexp_  

meta set command line
if specification
in specification

System variables
_meta_id_  
_meta_es_  
_meta_se_  
_meta_cil_  
_meta_ciu_  
_meta_studylabel_  
_meta_studysize_  

study ID variable
variable containing effect sizes
variable containing effect-size standard errors
variable containing lower bounds of CIs for effect sizes
variable containing upper bounds of CIs for effect sizes
string variable containing study labels
variable containing total sample size per study

References


Also see

[META] meta data — Declare meta-analysis data
[META] meta esize — Compute effect sizes and declare meta-analysis data
[META] meta update — Update, describe, and clear meta-analysis settings
[META] meta — Introduction to meta
[META] Glossary
[META] Intro — Introduction to meta-analysis