

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

Description
Options
Also see

Quick start
Remarks and examples

Menu
Stored results

Syntax
References

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(HR)` in the output

```
meta set es se, common studylabel(studylab) eslabel("log(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.

<i>options</i>	Description
Model	
<code>random</code> [<i>(remethod)</i>]	random-effects meta-analysis; default is <code>random(rem1)</code>
<code>common</code>	common-effect meta-analysis; implies inverse-variance method
<code>fixed</code>	fixed-effects meta-analysis; implies inverse-variance method
Options	
<code>studylabel</code> (<i>varname</i>)	variable to be used to label studies in all meta-analysis output
<code>studysize</code> (<i>varname</i>)	total sample size per study
<code>eslabel</code> (<i>string</i>)	effect-size label to be used in all meta-analysis output; default is <code>eslabel(Effect Size)</code>
<code>level</code> (#)	confidence level for all subsequent meta-analysis commands
<code>[no]metashow</code>	display or suppress meta settings with other <code>meta</code> commands

<i>remethod</i>	Description
<code>rem1</code>	restricted maximum likelihood; the default
<code>mle</code>	maximum likelihood
<code>ebayes</code>	empirical Bayes
<code>dlaird</code>	DerSimonian–Laird
<code>sjonkman</code>	Sidik–Jonkman
<code>hedges</code>	Hedges
<code>hschmidt</code>	Hunter–Schmidt

Options

Main

`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.

Model

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 τ^2 . *remethod* is one of `reml`, `mle`, `ebayes`, `dlaird`, `sjonkman`, `hedges`, or `hschmidt`. `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 τ^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 τ^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 τ^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 `dlaird`. Method `ebayes` produces a nonnegative estimate of τ^2 and requires iteration.

`dlaird` specifies that the DerSimonian–Laird method (DerSimonian and Laird 1986) be used to estimate τ^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 τ^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 τ^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 τ^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 τ^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.

Options

`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)
. describe
```

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

```
obs:           10           Generic effect sizes; fictional
                        data
vars:           9           19 Apr 2019 01:28
```

variable name	storage type	display format	value label	variable label
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
cil90	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`.

```
. meta set es se
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: REML
```

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

Study	Effect Size	[95% Conf. Interval]		% Weight
Study 1	1.480	-0.352	3.311	2.30
Study 2	0.999	-0.933	2.931	2.07
Study 3	1.272	0.427	2.117	10.15
Study 4	1.001	0.750	1.252	63.77
Study 5	1.179	-0.527	2.884	2.65
Study 6	1.939	0.427	3.452	3.35
Study 7	2.377	1.005	3.750	4.05
Study 8	0.694	-0.569	1.956	4.75
Study 9	1.099	-0.147	2.345	4.88
Study 10	1.805	-0.151	3.761	2.02
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
```

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

Meta-analysis summary
Random-effects model
Method: REML

Number of studies = 10
Heterogeneity:
  tau2 = 0.0157
  I2 (%) = 5.30
  H2 = 1.06
```

Study	Effect Size	[95% Conf. Interval]		% Weight
Study 1	1.480	-0.352	3.311	2.30
Study 2	0.999	-0.933	2.931	2.07
Study 3	1.272	0.427	2.117	10.15
Study 4	1.001	0.750	1.252	63.77
Study 5	1.179	-0.527	2.884	2.65
Study 6	1.939	0.427	3.452	3.35
Study 7	2.377	1.005	3.750	4.05
Study 8	0.694	-0.569	1.956	4.75
Study 9	1.099	-0.147	2.345	4.88
Study 10	1.805	-0.151	3.761	2.02
theta	1.138	0.857	1.418	

```
Test of theta = 0: z = 7.95
Test of homogeneity: Q = chi2(9) = 6.34

Prob > |z| = 0.0000
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)

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: [cil90, ciu90]
  User CI level: 90%, controlled by civarlevel()

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

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          Heterogeneity:
Method: REML                  tau2 = 0.0157
                              I2 (%) = 5.30
                              H2 = 1.06
```

Study	Effect Size	[95% Conf. Interval]		% Weight
Study 1	1.480	-0.352	3.311	2.30
Study 2	0.999	-0.933	2.931	2.07
Study 3	1.272	0.427	2.117	10.15
Study 4	1.001	0.750	1.252	63.77
Study 5	1.179	-0.527	2.884	2.65
Study 6	1.939	0.427	3.452	3.35
Study 7	2.377	1.005	3.750	4.05
Study 8	0.694	-0.569	1.956	4.75
Study 9	1.099	-0.147	2.345	4.88
Study 10	1.805	-0.151	3.761	2.02
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(studylabel) 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
Meta-analysis summary
Random-effects model
Method: REML
Number of studies = 10
Heterogeneity:
  tau2 = 0.0157
  I2 (%) = 5.30
  H2 = 1.06

```

Study	Mean Diff.	[90% Conf. Interval]	% Weight
Smith et al. (1984)	1.480	-0.057 3.016	2.30
Jones and Miller (1989)	0.999	-0.622 2.620	2.07
Johnson et al. (1991)	1.272	0.563 1.981	10.15
Brown et al. (1995)	1.001	0.790 1.211	63.77
Clark and Thomas (1998)	1.179	-0.252 2.610	2.65
Williams et al. (2003)	1.939	0.670 3.209	3.35
Davis and Wilson (2010)	2.377	1.226 3.529	4.05
Moore and Parker (2014)	0.694	-0.366 1.753	4.75
Miller et al. (2018)	1.099	0.053 2.144	4.88
Assaad et al. (2019)	1.805	0.164 3.446	2.02
theta	1.138	0.902 1.373	

```

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

<code>_dta[_meta_marker]</code>	“_meta_ds_1”
<code>_dta[_meta_K]</code>	number of studies in the meta-analysis
<code>_dta[_meta_studylabel]</code>	name of string variable containing study labels or <code>Generic</code>
<code>_dta[_meta_studysize]</code>	name of numeric variable containing study sizes, when <code>studysize()</code> specified
<code>_dta[_meta_estype]</code>	type of effect size; <code>Generic</code>
<code>_dta[_meta_eslabelopt]</code>	<code>eslabel(eslab)</code> , if specified
<code>_dta[_meta_eslabel]</code>	effect-size label from <code>eslabel()</code> ; default is <code>Effect Size</code>
<code>_dta[_meta_eslabeldb]</code>	effect-size label for dialog box
<code>_dta[_meta_esvar]</code>	name of effect-size variable
<code>_dta[_meta_esvardb]</code>	abbreviated name of effect-size variable for dialog box
<code>_dta[_meta_sevar]</code>	name of standard-error variable, if specified, or <code>_meta_se</code>
<code>_dta[_meta_cilvar]</code>	name of variable containing lower CI bounds, if specified, or <code>_meta_cil</code>
<code>_dta[_meta_ciuvar]</code>	name of variable containing upper CI bounds, if specified, or <code>_meta_ciu</code>
<code>_dta[_meta_civarlevel]</code>	confidence level associated with CI variables, if specified
<code>_dta[_meta_level]</code>	default confidence level for meta-analysis
<code>_dta[_meta_modellabel]</code>	meta-analysis model label: <code>Random-effects</code> , <code>Common-effect</code> , or <code>Fixed-effects</code>
<code>_dta[_meta_model]</code>	meta-analysis model: <code>random</code> , <code>common</code> , or <code>fixed</code>
<code>_dta[_meta_methodlabel]</code>	meta-analysis method label; varies by meta-analysis model
<code>_dta[_meta_method]</code>	meta-analysis method; varies by meta-analysis model
<code>_dta[_meta_randomopt]</code>	<code>random(remethod)</code> , if specified
<code>_dta[_meta_show]</code>	empty or <code>nometashow</code>
<code>_dta[_meta_datatype]</code>	data type; <code>Generic</code>
<code>_dta[_meta_datavars]</code>	variables specified with <code>meta set</code>

<code>_dta[_meta_setcmdline]</code>	meta set command line
<code>_dta[_meta_ifexp]</code>	if specification
<code>_dta[_meta_inexp]</code>	in specification

System variables

<code>_meta_id</code>	study ID variable
<code>_meta_es</code>	variable containing effect sizes
<code>_meta_se</code>	variable containing effect-size standard errors
<code>_meta_cil</code>	variable containing lower bounds of CIs for effect sizes
<code>_meta_ciu</code>	variable containing upper bounds of CIs for effect sizes
<code>_meta_studylabel</code>	string variable containing study labels
<code>_meta_studysize</code>	variable containing total sample size per study

References

- Berkey, C. S., D. C. Hoaglin, F. Mosteller, and G. A. Colditz. 1995. A random-effects regression model for meta-analysis. *Statistics in Medicine* 14: 395–411.
- DerSimonian, R., and N. M. Laird. 1986. Meta-analysis in clinical trials. *Controlled Clinical Trials* 7: 177–188.
- Hardy, R. J., and S. G. Thompson. 1996. A likelihood approach to meta-analysis with random effects. *Statistics in Medicine* 15: 619–629.
- Hedges, L. V. 1983. A random effects model for effect sizes. *Psychological Bulletin* 93: 388–395.
- Paule, R. C., and J. Mandel. 1982. Consensus values and weighting factors. *Journal of Research of the National Bureau of Standards* 87: 377–385.
- Raudenbush, S. W. 2009. Analyzing effect sizes: Random-effects models. In *The Handbook of Research Synthesis and Meta-Analysis*, ed. H. Cooper, L. V. Hedges, and J. C. Valentine, 2nd ed., 295–316. New York: Russell Sage Foundation.
- Schmidt, F. L., and J. E. Hunter. 2015. *Methods of Meta-Analysis: Correcting Error and Bias in Research Findings*. 3rd ed. Thousand Oaks, CA: SAGE.
- Sidik, K., and J. N. Jonkman. 2005. A note on variance estimation in random effects meta-regression. *Journal of Biopharmaceutical Statistics* 15: 823–838.
- Veroniki, A. A., D. Jackson, W. Viechtbauer, R. Bender, J. Bowden, G. Knapp, O. Kuss, J. P. T. Higgins, D. Langan, and G. Salanti. 2016. Methods to estimate the between-study variance and its uncertainty in meta-analysis. *Research Synthesis Methods* 7: 55–79.

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