

meta meregress — Multilevel mixed-effects meta-regression

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`meta meregress` performs multilevel meta-regression. You can think of multilevel meta-regression as an extension of [meta-regression](#), where effect sizes are nested within a higher grouping variable, such as district or region, and thus may be correlated. These groups may themselves be nested within another higher grouping variable, and so on. The dependencies among the observations within a group are accounted for by the inclusion of random effects at different levels of hierarchy.

If you wish to fit multilevel meta-analysis models with random intercepts only, see [\[META\] meta multilevel](#) for an alternative command with a simpler syntax.

`meta meregress` performs random-effects (RE) multilevel meta-regression with various covariance structures and estimation methods for the random effects, which include random intercepts and random coefficients. `meta meregress` is a stand-alone command in that it does not require you to declare your data as meta data using `meta set` or `meta esize`.

Quick start

Perform standard RE meta-analysis by expressing it as a two-level meta-analysis model of the effect-size `y` with random intercepts by `trial` and effect-size standard errors `se`

```
meta meregress y || trial:, essevariable(se)
```

As above, but perform an RE meta-regression on continuous moderator `x`

```
meta meregress y x || trial:, essevariable(se)
```

As above, but specify effect-size variances (`var`) instead of the effect-size standard errors

```
meta meregress y x || trial:, esvarvariable(var)
```

Perform a three-level meta-analysis of effect-size `y` with random intercepts by `region` and by `trial` nested within `region`

```
meta meregress y || region: || trial:, essevariable(se)
```

As above, but perform a three-level meta-regression on moderator `x`, add a random slope on `x` at the region level, and request the ML instead of the default REML estimation method

```
meta meregress y x || region: x || trial:, essevariable(se) mle
```

As above, but add a random slope on `x` at the trial-within-region level and specify an `exchangeable` covariance structure between the random slopes and intercepts at the trial-within-region level and an unstructured covariance structure between the random slopes and intercepts at the region level

```
meta meregress y x || region: x, covariance(unstructured) ///
|| trial: x, covariance(exchangeable) essevariable(se) mle
```

2 meta meregress — Multilevel mixed-effects meta-regression

Perform a three-level meta-regression of y on x_1 and x_2 with random slopes for x_1 and x_2 at the `region` level, and specify a custom covariance structure for the random effects at the `region` level

```
matrix A = (.5,.,.a .,1,. .a,.,1)
meta meregress y x1 x2 || region: x1 x2, covariance(custom A) ///
                    || trial:, essevariable(se)
```

Menu

Statistics > Meta-analysis

Syntax

```
meta meregress depvar fe_equation || re_equation [|| re_equation [...]],
    { essevariable(varname) | esvarvariable(varname) } [options]
```

where the syntax of *fe_equation* is

```
[indepvars] [if] [in] [, fe_options]
```

and the syntax of *re_equation* is

```
levelvar: [varlist] [, re_options]
```

levelvar is a variable identifying the group structure for the random effects at that level. A random intercept is included in each *re_equation* unless option `noconstant` is specified and a random coefficient (also known as a random slope) associated with each variable in *varlist* is also added to the model.

<i>fe_options</i>	Description
-------------------	-------------

<code>noconstant</code>	suppress constant term from the fixed-effects equation
<code>constraints</code> (<i>constraints</i>)	apply specified linear constraints

<i>re_options</i>	Description
-------------------	-------------

<code>covariance</code> (<i>vartype</i>)	variance-covariance structure of the random effects
<code>noconstant</code>	suppress constant term from the random-effects equation
<code>collinear</code>	keep collinear variables

<i>options</i>	Description
Model	
* <code>essevariable(<i>varname</i>)</code>	specify effect-size (sampling) standard errors
* <code>esvarvariable(<i>varname</i>)</code>	specify effect-size (sampling) variances
<code>reml</code>	fit model via restricted maximum likelihood; the default
<code>mle</code>	fit model via maximum likelihood
Reporting	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>stddeviations</code>	show random-effects parameter estimates as standard deviations and correlations; the default
<code>variance</code>	show random-effects parameter estimates as variances and covariances
<code>estmetric</code>	show parameter estimates as stored in <code>e(b)</code>
<code>nohomtest</code>	suppress output for homogeneity test
<code>norettable</code>	suppress random-effects table
<code>nofetable</code>	suppress fixed-effects table
<code>noheader</code>	suppress output header
<code>nogroup</code>	suppress table summarizing groups
<code>nocnsreport</code>	do not display constraints
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
EM options	
<code>emiterate(#)</code>	number of EM iterations; default is <code>emiterate(20)</code>
<code>emtolerance(#)</code>	EM convergence tolerance; default is <code>emtolerance(1e-10)</code>
<code>emonly</code>	fit model exclusively using EM
<code>emlog</code>	show EM iteration log
<code>emdots</code>	show EM iterations as dots
Maximization	
<code>maximize_options</code>	control the maximization process; seldom used
<code>coeflegend</code>	display legend instead of statistics

*Either `essevariable()` or `esvarvariable()` is required.

`indepvars` and `varlist` may contain factor variables; see [U] 11.4.3 **Factor variables**.

`collect` is allowed; see [U] 11.1.10 **Prefix commands**.

`coeflegend` does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

<i>vartype</i>	Description
<u>independent</u>	one unique standard-deviation parameter per random effect, all correlations 0; the default
<u>exchangeable</u>	equal standard deviations for random effects and one common pairwise correlation
<u>identity</u>	equal standard deviations for random effects; all correlations 0
<u>unstructured</u>	all standard deviations and correlations to be distinctly estimated
custom <i>matname</i>	custom matrix <i>matname</i> with fixed, free, and patterned standard deviations and correlations

Options

Model

`noconstant` suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any of or all the random-effects equations.

`covariance(vartype)` specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. *vartype* is one of the following: `independent`, `exchangeable`, `identity`, `unstructured`, or `custom`.

`independent` allows for a distinct standard deviation for each random effect within a random-effects equation and assumes that all correlations are 0. This is the default covariance structure.

`exchangeable` specifies one common standard deviation for all random effects and one common pairwise correlation.

`identity` is short for “multiple of the identity”; that is, all standard deviations are equal and all correlations are 0.

`unstructured` allows for all standard deviations and correlations to be distinct. If there are q random-effects terms, the unstructured covariance matrix will have $q(q+1)/2$ unique parameters.

`custom matname` specifies constraints for standard deviations (diagonal elements of *matname*) and correlations (off-diagonal elements of *matname*) of the random effects. Three types of specifications are allowed within *matname*:

1. A nonmissing value `#` that fixes the corresponding element at `#` during estimation.
2. One of `.a`, `.b`, etc., assigned to at least two diagonal or two off-diagonal elements to restrict the respective standard deviations or correlations to be the same during estimation.
3. A missing value `.` that allows the corresponding element to be freely estimated.

For example, assume that an *re_equation* in the model is `|| levelvar : x1 x2 x3` and therefore there are four random effects (one random intercept and three random slopes) at the *levelvar* level. Below, we describe the effect of specifying `covariance(custom matname)` with

$$matname = \begin{pmatrix} & x1 & x2 & x3 & _cons \\ 1.2 & & & & \\ 0.5 & .a & & & \\ .b & . & .a & & \\ .b & .c & .c & . & \end{pmatrix}$$

Let the u_j 's be the random slopes of x_j , $j \in \{1, 2, 3\}$ and u_0 be the random intercept. The above specification fixes the standard deviation of u_1 at 1.2 and the correlation between u_1 and u_2 at 0.5 during estimation. It also restricts the standard deviations of u_2 and u_3 to be equal (set equal to .a), the correlation between u_1 and u_3 to be the same as the correlation between u_1 and u_0 (both set equal to .b), and the correlation between u_2 and u_0 to be identical to the correlation between u_3 and u_0 (both set equal to .c). Furthermore, it allows the standard deviation of u_0 and the correlation between u_2 and u_3 to be freely estimated.

`essevariable(varname)` specifies a variable that stores the standard errors of the effect sizes in variable *varname*, also known as sampling standard errors. You must specify one of `essevariable()` or `esvarvariable()`.

`esvarvariable(varname)` specifies a variable that stores the variances of the effect sizes in variable *varname*, also known as sampling variances. You must specify one of `esvarvariable()` or `essevariable()`.

`reml` and `mle` specify the statistical method for fitting the model.

`reml`, the default, specifies that the model be fit using restricted maximum likelihood (REML), also known as residual maximum likelihood.

`mle` specifies that the model be fit using maximum likelihood (ML).

`constraints(constraints)`; see [R] [Estimation options](#).

Reporting

`level(#)`; see [R] [Estimation options](#).

`stddeviations`, the default, displays the random-effects parameter estimates as standard deviations and correlations.

`variance` displays the random-effects parameter estimates as variances and covariances.

`estmetric`; see [ME] [mixed](#).

`nohomtest` suppresses the homogeneity test based on the Q_M statistic from the output.

`noretabel`, `nofetable`, `noheader`, and `nogroup`; see [ME] [mixed](#).

`nocnsreport`; see [R] [Estimation options](#).

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] [Estimation options](#).

EM options

`emiterate(#)`, `emtolerance(#)`, `emonly`, `emlog`, and `emdots`; see [ME] [mixed](#).

Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, and `nonrtolerance`; see [R] [Maximize](#). Those that require special mention for meta meregress are listed below.

For the `technique()` option, the default is `technique(nr)`. The `bhhh` algorithm is not available.

`matsqrt`, the default, and `matlog`; see [ME] [mixed](#).

The following options are available with `meta meregress` but are not shown in the dialog box:

`collinear` specifies that `meta meregress` not omit collinear variables from the random-effects equation. Usually, there is no reason to leave collinear variables in place; in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the `collinear` option allows the estimation to take place with the random-effects equation intact.

`coeflegend`; see [R] [Estimation options](#).

Remarks and examples

[stata.com](http://www.stata.com)

Remarks are presented under the following headings:

Introduction

Standard meta-analysis as a two-level model

Three-level random-intercepts model

Three-level model with random slopes

Using meta meregress

Examples of using meta meregress

Example 1: Standard meta-analysis as a two-level model

Example 2: Three-level meta-analysis

Example 3: Assessing multilevel heterogeneity

Example 4: Likelihood-ratio tests and information criteria

Example 5: Three-level meta-regression with random slopes

Example 6: Random-effects covariance structures

Example 7: Sensitivity multilevel meta-regression

Introduction

Multilevel meta-regression is a statistical technique used to study the relationship between effect sizes and covariates, where effect sizes may be correlated because of the clustered or multilevel (hierarchical) structure of the data. The multilevel structure can arise, for example, when we consider a meta-analysis that explores the impact of a new teaching technique on math testing scores. Studies may be conducted in separate school districts with potentially multiple studies in each school district. Each study reports an effect size that quantifies the difference between the two groups of students (those who received the new teaching technique and those who did not), such as mean difference of testing scores between the two groups. We are interested not only in synthesizing the overall effect of the new teaching technique but also in assessing the variability (heterogeneity) among the effect sizes at the district level (level 3) and among the studies within each district (level 2, also known as the studies-within-district level).

Results of studies conducted within the same school district are more likely to be similar and thus dependent given that, for example, the students therein are exposed to the same socioeconomic factors. This dependence is usually accounted for by including random effects at various levels of hierarchy in the model. By properly accounting for the dependence among the effect sizes, we can produce more accurate inference compared with performing a standard meta-analysis that ignores the hierarchical structure and the dependence among the effect sizes.

The standard meta-analysis can be viewed as a two-level meta-analysis model where the subjects or participants within studies are the level-1 observations and studies (or more precisely effect sizes reported by the studies) are the level-2 observations. The within-study standard errors or variances are assumed known; see *Standard meta-analysis as a two-level model*.

In the school example above, studies are the level-2 observations. However, this is not always the case in multilevel meta-analysis applications. For example, we may have a four-level meta-analysis with runs (level 2) nested within experiments (level 3) nested within studies (level 4); see [McCurdy et al. \(2020\)](#) for another example. Here studies actually define level 4 and runs define level 2. Thus, the terms “within-study standard errors” and “within-study variances” may not always be appropriate to refer to the variability at the lowest level of hierarchy, which is described by the standard errors or variances of the effect sizes. In our four-level example, the terms “within-run standard errors” and “within-run variances” would be more appropriate. To avoid any confusion, we will use the terms “sampling standard errors” and “sampling variances” to refer, respectively, to the standard errors and the variances of the effect sizes.

Multilevel meta-regression differs from standard meta-regression in two major aspects. First, a hierarchical (grouping) structure is assumed to be present in the data, and it is a main interest to decompose the total heterogeneity among the effect sizes across the different levels of hierarchy. Second, random slopes for moderators may be included in the model at different grouping levels. Recall that a standard meta-regression model incorporates only random intercepts.

Multilevel meta-regression is analogous to a multilevel mixed-effects model ([Raudenbush and Bryk 2002](#)), which is used when individual data are available, but in multilevel meta-regression, the outcome of interest is an effect size. And, because we do not have individual participant data, there are no covariates that are recorded at the lowest observation level. Also, the sampling variances, the variability at the lowest level, are assumed to be known. Having the known sampling variance allows us to include random intercepts at level 2. However, to include random slopes at level 2, the data must include repeated measures at this level; in the example of schools at level 2, we would need multiple effect sizes for each school to include random slopes at the school level. At level 3 and higher, the data will naturally have multiple lower-level groups nested within higher-level groups, so both random intercepts and random slopes can be included.

The covariates in multilevel meta-regression are known as moderators. Examples of moderators include study publication year, study test environment, and drug administration method. For a comprehensive introduction to multilevel meta-regression, see [Goldstein et al. \(2000\)](#); [Thompson, Turner, and Warn \(2001\)](#); [Konstantopoulos \(2011\)](#); [Cheung \(2014\)](#); and [Sera et al. \(2019\)](#).

Standard meta-analysis as a two-level model

The standard RE meta-analysis model (see [\[META\] meta summarize](#)) may be viewed as a special two-level meta-analysis where the subjects or the within-study observations (level 1) are nested within studies (level 2). These levels are

$$\begin{aligned} \text{Level 1 (within studies): } \hat{\theta}_j &= \theta_j + \epsilon_j \\ \text{Level 2 (between studies): } \theta_j &= \theta + u_j \end{aligned} \tag{1}$$

where $\epsilon_j \sim N(0, \hat{\sigma}_j^2)$ and $u_j \sim N(0, \tau^2)$. Here $\hat{\sigma}_j^2$ is the sampling variance (effect-size variance) for the j th study, which is assumed known (it is assumed to be estimated with adequate accuracy within each study, hence the hat notation). τ^2 is the variance of the random effects (the u_j 's), also known as the between-study variance. The sampling errors (the ϵ_j 's) and the random effects (the u_j 's) are assumed to be independent. Similarly, the classical RE meta-regression (see [\[META\] meta regress](#)) can be obtained by incorporating moderators into (1) as follows:

Level 1 (within studies): $\hat{\theta}_j = \theta_j + \epsilon_j$

Level 2 (between studies): $\theta_j = \beta_0 + \beta_1 x_{1,j} + \dots + \beta_{p-1} x_{p-1,j} + u_j$
 $= \mathbf{x}_j \boldsymbol{\beta} + u_j$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_{p-1})'$ is a $p \times 1$ vector of unknown regression (fixed-effects) coefficients.

Three-level random-intercepts model

Next we will discuss extensions of (1) to higher levels of hierarchy. Given its prevalence in practice, we will start by mathematically describing the three-level random-intercepts meta-analysis model with a single observation per level-2 group (level-2 groups have no repeated measures). The model can be expressed as

Level 1 (within studies): $\hat{\theta}_{jk} = \theta_{jk} + \epsilon_{jk}$

Level 2: $\theta_{jk} = \theta_j + u_{jk}^{(2)}$ (2)

Level 3: $\theta_j = \theta + u_j^{(3)}$

where $j = 1, 2, \dots, M$, $k = 1, 2, \dots, m_j$, $u_j^{(3)} \sim N(0, \tau_3^2)$, $u_{jk}^{(2)} \sim N(0, \tau_2^2)$, and $\epsilon_{jk} \sim N(0, \hat{\sigma}_{jk}^2)$, with the $\hat{\sigma}_{jk}^2$'s being known as sampling variances (or more generally within-level-2 variances). The $u_j^{(3)}$'s, $u_{jk}^{(2)}$'s, and ϵ_{jk} 's are independent, and τ_3^2 and τ_2^2 are the random-effects variances at the third and second levels, respectively. Model (2) assumes that there is one effect-size observation per group at level 2. This is the most common setting in practice. For the general setting that accounts for multiple observations per group at level 2, see [Three-level model with random slopes](#).

In a single-equation notation, (2) can be written as

$$\hat{\theta}_{jk} = \theta + u_j^{(3)} + u_{jk}^{(2)} + \epsilon_{jk}$$

When we include a $1 \times p$ vector of moderators, $\mathbf{x}_{jk} = (1, x_{1,jk}, \dots, x_{p-1,jk})$, the three-level meta-analysis model described in (2) becomes a three-level meta-regression model

$$\begin{aligned} \hat{\theta}_{jk} &= \beta_0 + \beta_1 x_{1,jk} + \dots + \beta_{p-1} x_{p-1,jk} + u_j^{(3)} + u_{jk}^{(2)} + \epsilon_{jk} \\ &= \mathbf{x}_{jk} \boldsymbol{\beta} + u_j^{(3)} + u_{jk}^{(2)} + \epsilon_{jk} \end{aligned} \quad (3)$$

Model (3) includes only random intercepts. It does not include any random slopes for moderators \mathbf{x}_{jk} . The above model can be extended to more than three levels of hierarchy; see [Sera et al. \(2019\)](#) for details. If you would like to fit a model like (3) or its higher-level analogs, you can use the `meta multilevel` command, which has a simpler syntax than `meta meregress`.

Three-level model with random slopes

Incorporating random slopes at any level of hierarchy (other than level 1, where observations or individual participants are not available) requires repeated measures to be available at that level. For example, it is not possible to include random slopes at level 2 in meta-analysis for any of the moderators in (3) because there is one observation per group at that level. In that case, if you attempt to include random slopes at level 2, `meta meregress` will produce estimates of their standard deviations that are practically 0. If you are familiar with the concept of random slopes in the context of multilevel meta-analysis, then you may skip the rest of this section and go to [Using meta meregress](#).

Below, we modify the notation used in (3) to introduce a third subscript, r , that accounts for the repeated measures at level 2. Assume there are q_l random effects (1 random intercept and $q_l - 1$ random slopes) at level $l = 2$ and $l = 3$; random slopes may then be introduced into (3) by writing

$$\hat{\theta}_{jkr} = \mathbf{x}_{jkr} \boldsymbol{\beta} + \mathbf{z}_{jkr}^{(3)} \mathbf{u}_j^{(3)} + \mathbf{z}_{jkr}^{(2)} \mathbf{u}_{jk}^{(2)} + \epsilon_{jkr} \quad (4)$$

where $j = 1, 2, \dots, M$, $k = 1, 2, \dots, m_j$, and $r = 1, 2, \dots, m_{jk}$. The subscript r was not needed in (3), because m_{jk} was assumed to equal 1. Here $\mathbf{x}_{jkr} = (1, x_{1,jkr}, \dots, x_{p-1,jkr})$ is a $1 \times p$ vector of moderators associated with $\boldsymbol{\beta}$, and $\mathbf{z}_{jkr}^{(3)}$ is a $1 \times q_3$ vector of moderators associated with the level-3 $q_3 \times 1$ vector of random effects $\mathbf{u}_j^{(3)}$ (1 intercept and $q_3 - 1$ slopes), where $\mathbf{u}_j^{(3)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}^{(3)})$. Similarly, $\mathbf{z}_{jkr}^{(2)}$ is a $1 \times q_2$ vector of moderators associated with the level-2 (within-level-3) $q_2 \times 1$ vector of random effects $\mathbf{u}_{jk}^{(2)}$, where $\mathbf{u}_{jk}^{(2)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}^{(2)})$. The ϵ_{jkr} 's are the within-level-2 error terms following a $N(0, \hat{\sigma}_{jkr}^2)$ distribution. $\boldsymbol{\Sigma}^{(3)}$ and $\boldsymbol{\Sigma}^{(2)}$ are the random-effects covariance matrices at levels 3 and 2, respectively.

The above model can be extended to more than three levels of hierarchy with the possibility to include random slopes at any level $l > 1$; see [Sera et al. \(2019\)](#) for details.

Using meta meregress

`meta meregress` fits various multilevel meta-regression models. Suppose variable `es` records effect sizes and variable `se` records the sampling standard errors for effect sizes.

Standard meta-analysis model as a two-level model. The standard RE meta-analysis model can be expressed as a two-level meta-analysis model. Suppose variable `study` stores study IDs; we can then fit a standard RE meta-analysis model using

```
. meta meregress es || study:, essevariable(se)
```

Recall that in meta-analysis, the sampling standard errors are treated as known. We specify them in the `essevariable()` option. If you have variances instead, you can specify them in the `esvarvariable()` option. The above specification should produce the same results as if we had typed `meta set es se` followed by `meta summarize, nostudies`; see [\[META\] meta summarize](#) and [example 1](#).

Two-level meta-regression. Suppose we have two moderators, `x1` and `x2`. If we assume that the effects of moderators are constant across studies, we can fit a standard RE meta-regression as a two-level meta-regression without random coefficients (random slopes) for moderators:

```
. meta meregress es x1 x2 || study:, essevariable(se)
```

The above specification produces the same results as if we had typed `meta set es se` followed by `meta regress x1 x2`; see [\[META\] meta regress](#) and [example 1](#).

Alternatively, we can allow the effects of moderators to vary across studies by including random slopes for the moderators:

```
. meta meregress es x1 x2 || study: x1, essevariable(se)
```

Recall that this is possible only if there are multiple observations (effect sizes) per study; otherwise, the estimated standard deviations of the random slopes will be estimated as zeros. In other words, in the context of standard meta-regression (where one effect size per study is reported), the above specification will produce a zero estimate for the standard deviation of random slopes; see [Three-level model with random slopes](#) for more details.

We can include random slopes for all or a subset of moderators by specifying the desired subset in the random-effects equation (the `|| study:` equation in our example).

Three-level meta-analysis model. Suppose we have schools (level 2) and each school records repeated observations on effect sizes. Also, suppose that the schools are nested within districts (level 3). We can incorporate potential dependence among the effect sizes within schools and within districts by fitting a three-level meta-analysis model with `district` as the top (third) level:

```
. meta meregress es || district: || school:, essevariable(se)
```

Three-level meta-regression. Continuing with our three-level school data, if we also have moderators, say, `x1` and `x2`, we can incorporate them in our three-level meta-analysis model in various ways.

We can specify them only in the fixed-effects equation, assuming their effects do not vary across districts or schools within districts:

```
. meta meregress es x1 x2 || district: || school:, essevariable(se)
```

We can specify them in all equations to allow them to vary within all levels:

```
. meta meregress es x1 x2 || district: x1 x2 || school: x1 x2, essevariable(se)
```

Or, if there is only one effect size reported per school, then we can eliminate the random slopes from the school level:

```
. meta meregress es x1 x2 || district: x1 || school:, essevariable(se)
```

For illustration, in the above we included random slopes only for `x1`.

Three-level meta-regression with various covariance structures. In the presence of random slopes, we can specify various covariance structures to model the dependencies between random effects at a specific level. By default, the random effects are assumed to be independent. This default is chosen out of computational feasibility, in case the model includes many random slopes. In practice, you will often want to verify that this assumption is reasonable for your data. You can do this by specifying other covariance structures such as `exchangeable`, `unstructured`, or `custom matname` in the `covariance()` option. For instance, we now assume an unstructured (completely unrestricted) covariance for the random effects at the `district` level:

```
. meta meregress es x1 x2 || district: x1, covariance(unstructured)
  || school:, essevariable(se)
```

In some applications, you may need to fix or constrain some elements of the random-effects variance–covariance matrix. This is also useful to perform sensitivity analysis; see [example 7](#). You can do this by using the `custom matname` covariance structure.

Covariance structure `custom matname` provides a flexible way to restrict specific random-effects standard deviations and correlations during estimation while allowing the remaining parameters to be freely estimated. This option can be seen as a generalization of option `tau2()` in [\[META\] meta regress](#) and thus can be used to perform sensitivity analysis; see [covariance\(custom matname\)](#).

Similarly, we can build other models. With more levels, we can specify different covariance structures at different levels of hierarchy:

```
. meta meregress es x1 x2 || state: x1 x2, covariance(unstructured)
  || district: x2, covariance(exchangeable) || school: , essevariable(se)
```

By default, `meta meregress` uses the REML method to estimate model parameters. This method produces unbiased estimates of the random-effects covariance parameters by accounting for the loss of degrees of freedom from estimating the fixed-effects vector β . You can specify the `mle` option to instead estimate parameters using ML.

Examples of using meta meregress

Examples are presented under the following headings:

- Example 1: Standard meta-analysis as a two-level model*
- Example 2: Three-level meta-analysis*
- Example 3: Assessing multilevel heterogeneity*
- Example 4: Likelihood-ratio tests and information criteria*
- Example 5: Three-level meta-regression with random slopes*
- Example 6: Random-effects covariance structures*
- Example 7: Sensitivity multilevel meta-regression*

► Example 1: Standard meta-analysis as a two-level model

Recall the pupil IQ data (Raudenbush and Bryk 1985; Raudenbush 1984) described in *Effects of teacher expectancy on pupil IQ (pupiliq.dta)* of [META] **meta**. Here we will use its declared version (declared with `meta set`) to illustrate how to specify a standard meta-analysis model as a two-level random-intercepts model.

```
. use https://www.stata-press.com/data/r18/pupiliqset
(Effects of teacher expectancy on pupil IQ; set with -meta set-)
. meta query, short
-> meta set stdmdiff se , studylabel(study1bl) eslabel(Std. mean diff.)
Effect-size label: Std. mean diff.
Effect-size type: Generic
Effect size: stdmdiff
Std. err.: se
Model: Random effects
Method: REML
```

In these data, the effect sizes are standardized mean differences stored in variable `stdmdiff`, with their respective standard errors stored in variable `se`. To perform standard meta-analysis, we type the following, suppressing the individual study results for brevity:

```
. meta summarize, nostudies
Effect-size label: Std. mean diff.
Effect size: stdmdiff
Std. err.: se
Study label: study1bl

Meta-analysis summary
Random-effects model
Method: REML
Number of studies = 19
Heterogeneity:
tau2 = 0.0188
I2 (%) = 41.84
H2 = 1.72

theta: Overall Std. mean diff.
```

	Estimate	Std. err.	z	P> z	[95% conf. interval]
theta	.0836946	.0516536	1.62	0.105	-.0175447 .1849338

```
Test of homogeneity: Q = chi2(18) = 35.83 Prob > Q = 0.0074
```

The overall effect-size estimate is 0.0837 with the standard error of 0.052, and the estimated between-study variance `tau2` is 0.0188.

The standard meta-analysis model for this dataset can be expressed as

$$\text{stdmdiff}_j = \theta + u_j + \epsilon_j \quad (5)$$

with $u_j \sim N(0, \tau^2)$ and $\epsilon_j \sim N(0, \text{se}_j^2)$. This model can be fit using `meta meregress` as follows.

We specify the response variable and the fixed-effects portion of the model by typing the outcome variable (`stdmdiff`) and some independent variables (moderators) of interest (in this example, there are no moderators) after the command `meta meregress`. We then type `|| study:` to specify random effects at the study level. We did not specify any variables after the colon (`:`), because we wanted to incorporate only random intercepts; see [example 5](#) for random slopes. We also specify the variable containing the sampling standard errors using option `essevariable()`.

```
. meta meregress stdmdiff || study:, essevariable(se)
Performing EM optimization ...
Performing gradient-based optimization:
Iteration 0: Log restricted-likelihood = -33.419194 (not concave)
Iteration 1: Log restricted-likelihood = -10.213945 (not concave)
Iteration 2: Log restricted-likelihood = -3.8361073
Iteration 3: Log restricted-likelihood = -3.7393756
Iteration 4: Log restricted-likelihood = -3.7365412
Iteration 5: Log restricted-likelihood = -3.7365412
Computing standard errors ...
Multilevel REML meta-analysis
Group variable: study
Number of obs = 19
Number of groups = 19
Obs per group:
    min = 1
    avg = 1.0
    max = 1
Wald chi2(0) = .
Prob > chi2 = .
Log restricted-likelihood = -3.7365412
```

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_cons	.0836939	.0516531	1.62	0.105	-.0175444	.1849322

Test of homogeneity: $Q_M = \text{chi2}(18) = 35.83$ Prob > $Q_M = 0.0074$

Random-effects parameters	Estimate
study: Identity	
sd(_cons)	.1372184

The output shows information about the optimization algorithm, the iteration log, and the method (REML) used for estimating τ^2 . There are 19 observations (effect sizes) and 19 groups (studies) with one observation per group, which is the case for standard meta-analysis. The reported model Wald test is missing because we do not have moderators in our model.

The first table displays the fixed-effect parameter estimate from the two-level meta-analysis. Here the fixed-effect parameter is a constant term denoted by `_cons`, which represents θ in (5) and `theta` in the output from `meta summarize`. The estimate of θ is 0.0837 with a standard error of 0.052 and the 95% CI of $[-0.0175, 0.1849]$. The test of homogeneity, which tests that all effect sizes are equal, reports the Q_M statistic of 35.83 with a p -value of 0.0074. The second table shows the estimated value of τ (standard deviation of the random effects u_j 's) labeled as `sd(_cons)` in the output.

The results for the fixed-effect parameter are virtually identical. `meta summarize` reported an estimate of the variance of the random intercepts $\hat{\tau}^2 = 0.0188$, whereas `meta meregress` reports the standard deviation ($\hat{\tau} = 0.1372$) by default. We can display the variance by specifying the option `variance` on replay. We also use options `noheader` and `nofetable` to suppress the header and the fixed-effects table.

```
. meta meregress, variance noheader nofetable
```

Random-effects parameters	Estimate
study: Identity var(_cons)	.0188289

Alternatively, we could have used `estat sd`, `variance` to obtain the same output; see [\[META\] estat sd](#) and [example 6](#).

Similarly, we can fit a standard meta-regression model as a two-level random-intercepts regression model. First, we use `meta regress` ([\[META\] meta regress](#)) to fit a standard meta-regression model:

```
. meta regress weeks
```

```
Effect-size label: Std. mean diff.
```

```
Effect size: stdmdiff
```

```
Std. err.: se
```

```
Random-effects meta-regression
```

```
Method: REML
```

```
Number of obs = 19
```

```
Residual heterogeneity:
```

```
tau2 = .01117
```

```
I2 (%) = 29.36
```

```
H2 = 1.42
```

```
R-squared (%) = 40.70
```

```
Wald chi2(1) = 7.51
```

```
Prob > chi2 = 0.0061
```

_meta_es	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
weeks	-.0157453	.0057447	-2.74	0.006	-.0270046	-.0044859
_cons	.1941774	.0633563	3.06	0.002	.0700013	.3183535

```
Test of residual homogeneity: Q_res = chi2(17) = 27.66 Prob > Q_res = 0.0490
```

Next we fit this same model using `meta meregress`. We simply list the moderator (`weeks`) in the fixed-effects portion of the model after the outcome variable `stdmdiff`.

```
. meta meregress stdmdiff weeks || study:, essevariable(se) variance
Performing EM optimization ...
Performing gradient-based optimization:
Iteration 0: Log restricted-likelihood = -31.596287 (not concave)
Iteration 1: Log restricted-likelihood = -8.6658459 (not concave)
Iteration 2: Log restricted-likelihood = -1.1427859 (not concave)
Iteration 3: Log restricted-likelihood = -.71416907
Iteration 4: Log restricted-likelihood = -.71388211
Iteration 5: Log restricted-likelihood = -.71388211
Computing standard errors ...
Multilevel REML meta-regression
Group variable: study
Number of obs = 19
Number of groups = 19
Obs per group:
    min = 1
    avg = 1.0
    max = 1
Wald chi2(1) = 7.51
Prob > chi2 = 0.0061
Log restricted-likelihood = -.71388211
```

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
weeks	-.0157453	.0057447	-2.74	0.006	-.0270046	-.0044859
_cons	.1941769	.0633561	3.06	0.002	.0700012	.3183527

Test of homogeneity: $Q_M = \text{chi2}(17) = 27.66$ Prob > $Q_M = 0.0490$

Random-effects parameters	Estimate
study: Identity	
var(_cons)	.011166

The estimates for the fixed-effects coefficients (reported in the first table) and τ^2 (labeled `var(_cons)` in the second table above) are almost the same as from `meta regress`.

◀

► Example 2: Three-level meta-analysis

Consider a dataset from [Cooper et al. \(2003\)](#) on schools that modified their calendars without prolonging the school year. A version of this dataset was also analyzed by [Konstantopoulos \(2011\)](#) and will be used below. The dataset consists of 56 studies that were conducted in 11 school districts.

Some schools adopted modified calendars that feature shorter breaks more frequently throughout the year (for example, 12 weeks of school followed by 4 weeks off), as opposed to the traditional calendar with a longer summer break and shorter winter and spring breaks. The studies compared the academic achievement of students on a traditional calendar with those on a modified calendar. The effect size (`stdmdiff`) was the standardized mean difference with positive values indicating higher achievement, on average, in the group on the modified calendar. The standard error (`se`) of `stdmdiff` was also reported by each study. Let's first describe our dataset:

```

. use https://www.stata-press.com/data/r18/schoolcal, clear
(Effect of modified school calendar on student achievement)
. describe
Contains data from https://www.stata-press.com/data/r18/schoolcal.dta
Observations:          56          Effect of modified school
                                calendar on student achievement
Variables:             8           19 Jan 2023 21:44
                                (_dta has notes)

```

Variable name	Storage type	Display format	Value label	Variable label
district	int	%12.0g		District ID
school	byte	%9.0g		School ID
study	byte	%12.0g		Study ID
stdmdiff	double	%10.0g		Standardized difference in means of achievement test scores
var	double	%10.0g		Within-study variance of stdmdiff
year	int	%12.0g		Year of the study
se	double	%10.0g		Within-study standard-error of stdmdiff
year_c	byte	%9.0g		Year of the study centered around 1990

Sorted by: district

Because the schools are nested within districts, we fit a three-level random-intercepts model. This model can also be fit using command `meta multilevel`; see [example 1](#) of [\[META\] meta multilevel](#). The model can be expressed as

$$\text{stdmdiff}_{jk} = \theta + u_j^{(3)} + u_{jk}^{(2)} + \epsilon_{jk} \quad (6)$$

with $u_j^{(3)} \sim N(0, \tau_3^2)$, $u_{jk}^{(2)} \sim N(0, \tau_2^2)$, and $\epsilon_{jk} \sim N(0, \text{se}_{jk}^2)$. Here there is one observation (effect size) reported per school (level-2 group). Fitting a three-level model requires that you specify two random-effects equations: one for level 3 (identified by variable `district`) and one for level 2 (identified by variable `school`). This model can be fit using `meta meregress` as follows:

```
. meta meregress stdmdiff || district: || school:, essevariable(se)
Performing EM optimization ...
Performing gradient-based optimization:
Iteration 0: Log restricted-likelihood = -104.8525 (not concave)
Iteration 1: Log restricted-likelihood = -49.423271 (not concave)
Iteration 2: Log restricted-likelihood = -25.793841 (not concave)
Iteration 3: Log restricted-likelihood = -21.310018
Iteration 4: Log restricted-likelihood = -9.1236345
Iteration 5: Log restricted-likelihood = -8.2625776
Iteration 6: Log restricted-likelihood = -7.9588561
Iteration 7: Log restricted-likelihood = -7.9587239
Iteration 8: Log restricted-likelihood = -7.9587239
Computing standard errors ...
```

Multilevel REML meta-analysis Number of obs = 56

Grouping information

Group variable	No. of groups	Observations per group		
		Minimum	Average	Maximum
district	11	3	5.1	11
school	56	1	1.0	1

Log restricted-likelihood = -7.9587239 Wald chi2(0) = .
Prob > chi2 = .

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
_cons	.1847132	.0845559	2.18	0.029	.0189866	.3504397

Test of homogeneity: Q_M = chi2(55) = 578.86 Prob > Q_M = 0.0000

Random-effects parameters	Estimate
district: Identity	
sd(_cons)	.2550724
school: Identity	
sd(_cons)	.1809324

We first store the results of the model so we can use them later in [example 4](#) to perform likelihood-ratio tests.

```
. estimates store main_model
```

As in [example 1](#), our fixed-effects equation contains only the dependent variable (effect sizes `stdmdiff`). But we have two random-effects equations. The first represents random intercepts [the $u_j^{(3)}$'s in (6)] at the district level (level 3), and the second represents random intercepts [the $u_{jk}^{(2)}$'s in (6)] at the school level (level 2). The order in which these are specified (from left to right) is important—`meta meregress` assumes that `school` is nested within `district`. Below, we describe each portion of the output in detail.

The output first displays information about the optimization, including an iteration log. The top of the header shows the method (REML) used for estimation and also displays the total number of observations, which is 56 in our example.

The information on groups at different levels of hierarchy is displayed as a table with one row for each grouping (level of hierarchy). For example, there are 11 groups (districts) at the district

level. Each group contains somewhere between 3 to 11 level-2 groups (schools). You can suppress this table with the `nogroup` or the `noheader` option, which will also suppress the rest of the header.

The second table displays the fixed-effects coefficients. In our example, there is only an intercept corresponding to the term θ in (6). The value of $\hat{\theta}$ is 0.185 with a 95% CI of [0.019, 0.35]. This means that, on average, students following the modified school calendar achieved higher scores than those who did not.

The third table displays the random-effects parameters, traditionally known as variance components in the context of multilevel or mixed-effects models. The variance-component estimates are now organized and labeled according to each level. By default, `meta meregress` reports standard deviations of the random intercepts (and correlations if they existed in the model) at each level. But you can instead specify the `variance` option to report variances (and covariances if they existed in the model). We have $\hat{\tau}_3 = 0.255$ and $\hat{\tau}_2 = 0.181$. These values are the building blocks for assessing heterogeneity across different hierarchical levels and are typically interpreted in that context; see [example 3](#) and [Higgins–Thompson heterogeneity statistics](#) in *Methods and formulas* in [META] `estat heterogeneity (me)` for details. In general, the higher the value of τ_l , the more heterogeneity is expected among the groups within level l .

◀

► Example 3: Assessing multilevel heterogeneity

Continuing with [example 2](#), let's use the postestimation command `estat heterogeneity` to quantify the multilevel heterogeneity among the effect sizes captured by the three-level meta-analysis model.

```
. estat heterogeneity
Method: Cochran
Joint:
  I2 (%) = 90.50
Method: Higgins-Thompson
district:
  I2 (%) = 63.32
school:
  I2 (%) = 31.86
Total:
  I2 (%) = 95.19
```

Cochran's I_Q^2 quantifies the amount of heterogeneity jointly for all levels of hierarchy. It is a direct extension to the multilevel setting of the classical I^2 statistic based on the DerSimonian–Laird method and thus has the same interpretation. For instance, $I_Q^2 = 90.50\%$ means that 90.50% of the variability among the effect sizes is due to true heterogeneity in our data as opposed to the sampling variability. See [Heterogeneity measures](#) in *Methods and formulas* in [META] `meta summarize` and [Residual heterogeneity measures](#) in *Methods and formulas* in [META] `meta regress` for details.

The value of the Cochran statistic is the same for all multilevel models with the same fixed-effects structure. This is because its computation is based on the Cochran multivariate Q statistic, which is calculated based only on the fixed-effects model; see [Cochran heterogeneity statistic](#) in *Methods and formulas* in [META] `estat heterogeneity (me)` for details.

Unlike the Cochran I_Q^2 statistic, the multilevel Higgins–Thompson I^2 statistics (Nakagawa and Santos 2012) provide ways to assess the contribution of each level of hierarchy to the total heterogeneity, in addition to their joint contribution. For example, between-schools heterogeneity or heterogeneity within districts (level-2 heterogeneity) is the lowest, accounting for about 32% of the total variation in

our data, whereas between-districts heterogeneity (level-3 heterogeneity) accounts for about 63% of the total variation. This is a direct consequence of the estimate of τ_3^2 being greater than that of τ_2^2 in [example 2](#). See [Higgins–Thompson heterogeneity statistics](#) in [Methods and formulas](#) in [META] [estat heterogeneity \(me\)](#) for details.

◀

▷ Example 4: Likelihood-ratio tests and information criteria

Suppose we wish to test whether there is a nonnegligible amount of heterogeneity within districts (that is, heterogeneity between the schools within a district). This amounts to testing $H_0: \tau_2^2 = 0$. We need to fit a model with $\tau_2^2 = 0$ and compare it with the model from [example 2](#). This is a two-level model with `district` as the second level of hierarchy (we eliminate the `school` level). We fit this model and store its results under the name `school_effect`. Recall that we had already saved our results for the three-level model in [example 2](#) under the name `main_model`. So we can use the `lrtest` command to conduct a likelihood-ratio test of our H_0 .

```
. quietly meta meregress stdmdiff || district: , essevariable(se)
. estimates store school_effect
. lrtest main_model school_effect

Likelihood-ratio test
Assumption: school_effect nested within main_model

LR chi2(1) = 48.52
Prob > chi2 = 0.0000

Note: The reported degrees of freedom assumes the null hypothesis is not on
the boundary of the parameter space. If this is not true, then the
reported test is conservative.

Note: LR tests based on REML are valid only when the fixed-effects
specification is identical for both models.
```

Because the null hypothesis value of τ_2^2 is at the boundary of the parameter space, the asymptotic distribution of the test statistic is a mixture of the χ_0^2 (a point mass at zero) and χ_1^2 distributions ([Verbeke and Molenberghs 2000](#); [Self and Liang 1987](#); and [Gutierrez, Carter, and Drukker 2001](#)), with each having an equal weight of 0.5. To elaborate on the first note reported by `lrtest`, the exact p -value can therefore be computed as

$$p = 0.5 \times P(\chi_0^2 > 48.52) + 0.5 \times P(\chi_1^2 > 48.52) = 0.5 \times P(\chi_1^2 > 48.52)$$

which is half of what is reported above. The second equality holds because the χ^2 distribution with zero degrees of freedom, χ_0^2 , places all probability mass at zero, and therefore $0.5 \times P(\chi_0^2 > 48.52) = 0$. This updated p -value computation does not affect our conclusion regarding the test result, which is that we reject the hypothesis that schools are homogeneous within districts.

Similarly, we may also wish to test whether there is a nonnegligible amount of heterogeneity between districts, which amounts to testing $H_0: \tau_3^2 = 0$. This is equivalent to fitting a standard RE meta-analysis where all 56 effect sizes are assumed independent. Hence, we use variable `study` as the grouping level in our model specification. Had we used `school`, the model would have clustered our 56 effect sizes into 11 groups, which would violate the independence assumption.

```

. quietly meta meregress stdmdiff || study: , essevariable(se)
. estimates store dist_effect
. lrtest main_model dist_effect
Likelihood-ratio test
Assumption: dist_effect nested within main_model
LR chi2(1) = 17.77
Prob > chi2 = 0.0000
Note: The reported degrees of freedom assumes the null hypothesis is not on
the boundary of the parameter space. If this is not true, then the
reported test is conservative.
Note: LR tests based on REML are valid only when the fixed-effects
specification is identical for both models.

```

The results of the test provide strong evidence that there is significant between-districts heterogeneity. Similar discussion applies to the computation of the exact p -value as above.

We can compare our models using information criteria by using the `estimates stats` command. We use option `all` to request AICc and CAIC in addition to the default AIC and BIC. We also use option `n()` to use $n - p = 55$ instead of $n = 56$ as the number of observations in the computation of BIC, AICc, and CAIC because our models used REML estimation.

```

. estimates stats main_model dist_effect school_effect, all n(55)
Information criteria

```

Model	N	ll(null)	ll(model)	df
main_model	55	.	-7.958724	3
dist_effect	55	.	-16.8455	2
school_eff~t	55	.	-32.21648	2

Model	AIC	BIC	AICc	CAIC
main_model	21.91745	27.93945	22.38804	30.93945
dist_effect	37.691	41.70566	37.92177	43.70566
school_eff~t	68.43295	72.44762	68.66372	74.44762

```

Legend: AIC is Akaike's information criterion.
BIC is Bayesian information criterion.
AICc is corrected Akaike's information criterion.
CAIC is consistent Akaike's information criterion.

```

All measures of information criteria favor the three-level model `main_model`.

◀

► Example 5: Three-level meta-regression with random slopes

For illustration purposes, we will use variable `year_c` to conduct a three-level meta-regression and include random slopes (corresponding to variable `year_c`) at the `district` level. We will not include random slopes at the `school` level, because there is only one observation (effect size) per school; otherwise, we will get an estimate that is practically zero for the standard deviation of the random slope of `year_c` at the `school` level; see *Three-level model with random slopes*. The model can be described as follows:

$$\text{stdmdiff}_{jk} = \beta_0 + \beta_1 \text{year_c}_{jk} + u_{0j}^{(3)} + u_{1j}^{(3)} \text{year_c}_{jk} + u_{jk}^{(2)} + \epsilon_{jk} \quad (7)$$

with $(u_{0j}^{(3)}, u_{1j}^{(3)})' \sim N(\mathbf{0}, \Sigma^{(3)})$, $u_{jk}^{(2)} \sim N(0, \tau_2^2)$, and $\epsilon_{jk} \sim N(0, \text{var}_{jk})$. By default, the 2×2 matrix $\Sigma^{(3)}$ is assumed diagonal, which means that the $u_{0j}^{(3)}$'s and $u_{1j}^{(3)}$'s are assumed independent. Other covariance structures can be specified with the `covariance()` option; see [example 6](#).

```
. meta meregress stdmdiff year_c || district: year_c || school;,
> esvarvariable(var)
```

Performing EM optimization ...

Performing gradient-based optimization:

```
Iteration 0: Log restricted-likelihood = -101.95646 (not concave)
Iteration 1: Log restricted-likelihood = -94.506522 (not concave)
Iteration 2: Log restricted-likelihood = -27.473266 (not concave)
Iteration 3: Log restricted-likelihood = -9.8063483
Iteration 4: Log restricted-likelihood = -7.2135276
Iteration 5: Log restricted-likelihood = -7.210109 (not concave)
Iteration 6: Log restricted-likelihood = -7.2100808 (not concave)
Iteration 7: Log restricted-likelihood = -7.2100609 (not concave)
Iteration 8: Log restricted-likelihood = -7.2098538
Iteration 9: Log restricted-likelihood = -7.2095961
Iteration 10: Log restricted-likelihood = -7.2095347
Iteration 11: Log restricted-likelihood = -7.2095302
Iteration 12: Log restricted-likelihood = -7.2095301
```

Computing standard errors ...

Multilevel REML meta-regression Number of obs = 56

Grouping information

Group variable	No. of groups	Observations per group		
		Minimum	Average	Maximum
district	11	3	5.1	11
school	56	1	1.0	1

Log restricted-likelihood = -7.2095301 Wald chi2(1) = 0.55
Prob > chi2 = 0.4577

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
year_c	.0096026	.0129302	0.74	0.458	-.0157402	.0349453
_cons	.1609574	.0823083	1.96	0.051	-.0003639	.3222787

Test of homogeneity: Q_M = chi2(54) = 550.26 Prob > Q_M = 0.0000

Random-effects parameters	Estimate
district: Independent	
sd(year_c)	.033531
sd(_cons)	.0643482
school: Identity	
sd(_cons)	.1808125

The estimate of the regression coefficient of variable `year_c` is 0.010 with a 95% CI of $[-0.016, 0.035]$. We do not see any evidence for the association between `stdmdiff` and `year_c` ($p = 0.458$). The estimates of the standard deviations of $u_{1j}^{(3)}$ and $u_{0j}^{(3)}$ (at the district level) are labeled in the output as `sd(year_c)` and `sd(_cons)` and are estimated to be 0.034 and 0.064, respectively. These values are the estimates of the square root of the diagonal elements of $\Sigma^{(3)}$. The covariance structure at the district level is labeled as `Independent`, which is the default

assumption. You may display the 2×2 matrix $\Sigma^{(3)}$ using the `estat recovariance` command; see [example 6](#). The estimate of τ_2 is 0.181.

◀

Although `year_c` did not explain the heterogeneity, we continue to include it as a moderator in our subsequent examples ([example 6](#) and [example 7](#)) for illustration purposes.

▶ Example 6: Random-effects covariance structures

Continuing with [example 5](#), we will explore different random-effects covariance structures for $\Sigma^{(3)}$ instead of the default independent structure. The default independent covariance structure is chosen out of computational feasibility. In multilevel modeling, it is important to start with an unrestricted covariance first, whenever feasible. It is also important to have meaningful baseline values for the moderators to make variance components interpretable. Here we include `year_c`, which is centered on 1990, instead of `year`, so that the intercept can be interpreted as the expected value in 1990 and the variance components can also be interpreted relative to this year. Let's specify the `covariance(unstructured)` option first. This assumes that all random effects have distinct standard deviations and correlations. We suppress the header and the iteration log and display results with 3 decimal points using the `noheader`, `nolog`, and `cformat(%9.3f)` options, which we store in the local macro 'options' for syntactical convenience.

```
. local options noheader nolog cformat(%9.3f)
. meta meregress stdmdiff year_c || district: year_c,
> covariance(unstructured) || school:, esvarvariable(var) 'options'
```

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
year_c	0.007	0.010	0.71	0.479	-0.013	0.028
_cons	0.160	0.076	2.12	0.034	0.012	0.308

Test of homogeneity: Q_M = chi2(54) = 550.26

Prob > Q_M = 0.0000

Random-effects parameters	Estimate
district: Unstructured	
sd(year_c)	0.028
sd(_cons)	0.082
corr(year_c, _cons)	1.000
school: Identity	
sd(_cons)	0.180

The random-effects covariance structure at the `district` level is now labeled `Unstructured:`. The correlation between the random slope and the random intercept is labeled as `corr(year_c, _cons)`. The estimated correlation value is 1 because, as we mentioned in [example 5](#), variable `year_c` did not explain any heterogeneity and was included here for illustration purposes only.

Instead of specifying one of the standard covariance structures (`independent`, `identity`, `exchangeable`, or `unstructured`), you may request a custom covariance structure where you can fix specific standard deviations or correlations while allowing others to be estimated. For example, the following matrix `A` fixes the correlation between $u_{0j}^{(3)}$ and $u_{1j}^{(3)}$ at 0.5 and allows for their standard deviations to be estimated from the data. See `covariance(custom matname)` for details.

```
. matrix A = (. ,.5 \ .5 ,.)
. meta meregress stdmdiff year_c || district: year_c, covariance(custom A)
> || school:, esvarvariable(var) 'options'
```

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
year_c	0.007	0.011	0.67	0.500	-0.014	0.028
_cons	0.170	0.082	2.08	0.038	0.010	0.330

Test of homogeneity: $Q_M = \text{chi2}(54) = 550.26$ Prob > $Q_M = 0.0000$

Random-effects parameters	Estimate
district: Custom	
sd(year_c)	0.026
sd(_cons)	0.116
corr(year_c,_cons)	0.500*
school: Identity	
sd(_cons)	0.180

(*) fixed during estimation

Note the asterisk that is appended next to the `corr(year_c, _cons)` value to emphasize that it was fixed during estimation.

You may additionally wish to constrain the two standard deviations of the random intercept and random slope to be the same (both specified equal to `.a`):

```
. matrix B = (.a ,.5 \ .5 ,.a)
. meta meregress stdmdiff year_c || district: year_c, covariance(custom B)
> || school:, esvarvariable(var) 'options'
```

stdmdiff	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
year_c	0.010	0.012	0.79	0.427	-0.014	0.034
_cons	0.154	0.076	2.02	0.043	0.005	0.304

Test of homogeneity: $Q_M = \text{chi2}(54) = 550.26$ Prob > $Q_M = 0.0000$

Random-effects parameters	Estimate
district: Custom	
sd(year_c _cons)	0.033
corr(year_c,_cons)	0.500*
school: Identity	
sd(_cons)	0.181

(*) fixed during estimation

We can display the random-effects covariance matrices $\Sigma^{(3)}$ (at the district level) and $\Sigma^{(2)}$ (at the school level), which is a scalar in our example, using the `estat recovariance` command ([META] [estat recovariance](#)). This is particularly useful if we specify a complicated custom covariance structure in our model using the `covariance(custom matname)` option (think 3×3 or larger covariance matrices).

```
. estat recovariance
Random-effects covariance matrix for level district
```

	year_c	_cons
year_c	.0010852	
_cons	.0005426	.0010852

```
Fixed parameter: corr(year_c,_cons)=.5.
Random-effects covariance matrix for level school
```

	_cons
_cons	.0326401

To see the corresponding correlation matrix, specify the `correlation` option.

You may also use `estat sd` (`[META] estat sd`) to display the variance-components parameters as variances and covariances (instead of the default standard deviations and correlations). This will also group together any parameters that were constrained to be the same.

```
. estat sd, variance
```

Random-effects parameters	Estimate
district: Custom	
var(year_c _cons)	.0010852
cov(year_c,_cons)	.0005426
school: Identity	
var(_cons)	.0326401

```
Note: corr(year_c,_cons)=.5 at district
level fixed during estimation.
```

◀

► Example 7: Sensitivity multilevel meta-regression

It is quite common in multilevel meta-regression to produce unstable estimates, especially when the number of observations is small relative to the number of parameters to be estimated. In this case, our goal may shift toward assessing the impact of different magnitudes of random-effects covariance parameters on the estimates of regression coefficients to evaluate the robustness of our results.

Continuing with (7) from [example 5](#), we can investigate the effect of no correlation, moderate correlation (0.4), and high correlation (0.8) between the random intercepts (the $u_{0j}^{(3)}$'s) and the random slopes (the $u_{1j}^{(3)}$'s) at the `district` level on the regression coefficient estimates. We will allow for the random-effects standard deviations to be estimated from the data. Thus, our fixed custom random-effects covariance matrices for the three scenarios are

```
. matrix Sigma1 = (.,0\0,.)
. matrix Sigma2 = (.,.4\0.4,.)
. matrix Sigma3 = (.,.8\0.8,.)
```

We fit the first model using the correlations of 0 and store the estimation results as `corr0`.

```
. quietly meta meregress stdmdiff year_c
> || district: year_c, covariance(custom Sigma1)
> || school:, esvarvariable(var)
. estimates store corr0
```

Next we fit the model with correlations of 0.4 and store results as `corr4` and the model with correlations of 0.8 and store results as `corr8`. For brevity, we suppressed the output from all commands by running them quietly.

```
. quietly meta meregress stdmdiff year_c
> || district: year_c, covariance(custom Sigma2)
> || school:, esvarvariable(var)
. estimates store corr4

. quietly meta meregress stdmdiff year_c
> || district: year_c, covariance(custom Sigma3)
> || school:, esvarvariable(var)
. estimates store corr8
```

We compare the estimates side by side by using `estimates table`:

```
. estimates table corr0 corr4 corr8,
> keep(stdmdiff:year_c stdmdiff:_cons) b(%8.3f) se(%8.3f)
```

Variable	corr0	corr4	corr8
year_c	0.006	0.007	0.007
	0.011	0.011	0.011
_cons	0.181	0.172	0.164
	0.090	0.083	0.078

Legend: b/se

As the correlation between the random intercepts and the random slopes at the `district` level increases, the coefficient estimate for `_cons` decreases. Also, the estimate becomes more precise (has a smaller standard error) as the correlation increases. Note also how the various magnitudes of correlations had little to no impact on the estimation of `year_c` (all values are near 0) because, as we saw in [example 5](#), variable `year_c` did not explain any heterogeneity and should have been excluded from the model.

◀

Stored results

`meta meregress` stores the following in `e()`:

Scalars

<code>e(N)</code>	total number of observations
<code>e(k)</code>	number of parameters
<code>e(k_f)</code>	number of fixed-effects parameters
<code>e(k_r)</code>	number of random-effects parameters
<code>e(k_rs)</code>	number of variances
<code>e(k_rc)</code>	number of covariances
<code>e(ll)</code>	log (restricted) likelihood
<code>e(rank)</code>	rank of $e(V)$
<code>e(ic)</code>	number of iterations
<code>e(df_m)</code>	model degrees of freedom
<code>e(chi2)</code>	model χ^2 Wald test statistic
<code>e(p)</code>	p -value for model test
<code>e(Q_M)</code>	multilevel Cochran Q_M residual homogeneity test statistic
<code>e(df_Q_M)</code>	degrees of freedom for residual homogeneity test
<code>e(p_Q_M)</code>	p -value for residual homogeneity test
<code>e(converged)</code>	1 if converged, 0 otherwise

Macros

<code>e(cmd)</code>	<code>meta meregress</code>
<code>e(cmdline)</code>	command as typed
<code>e(method)</code>	REML or ML
<code>e(title)</code>	title in estimation output
<code>e(chi2type)</code>	Wald; type of model χ^2 test
<code>e(depvar)</code>	name of dependent variable
<code>e(ivars)</code>	grouping variables
<code>e(indepvars)</code>	names of independent variables (moderators)
<code>e(esvarvariable)</code>	variable containing sampling variances (when <code>esvarvariable()</code> is specified)
<code>e(essevariable)</code>	variable containing sampling standard errors (when <code>essevariable()</code> is specified)
<code>e(redim)</code>	random-effects dimensions
<code>e(vartypes)</code>	variance-structure types
<code>e(revars)</code>	random-effects covariates
<code>e(technique)</code>	maximization technique
<code>e(datasignature)</code>	the checksum
<code>e(datasignaturevars)</code>	variables used in calculation of checksum
<code>e(emonly)</code>	emonly, if specified
<code>e(ml_method)</code>	type of ml method
<code>e(opt)</code>	type of optimization
<code>e(optmetric)</code>	<code>matsqrt</code> or <code>matlog</code> ; random-effects matrix parameterization
<code>e(properties)</code>	b V
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance-covariance matrix of the estimators
<code>e(cov_cust_#)</code>	custom random-effects covariance matrix (when <code>covariance(custom matname)</code> is specified)
<code>e(Cns)</code>	constraints matrix
<code>e(N_g)</code>	group counts
<code>e(g_min)</code>	group-size minimums
<code>e(g_avg)</code>	group-size averages
<code>e(g_max)</code>	group-size maximums

Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices

<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals
-----------------------	---

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r`-class command is run after the estimation command.

When the `esvarvariable()` option is specified, `meta meregress` creates a system variable, `_meta_mereg_se`, that contains the sampling standard errors.

Methods and formulas

Methods and formulas are presented under the following headings:

Three-level meta-regression

Methods for estimating $\Sigma^{(2)}$ and $\Sigma^{(3)}$

Random-effects covariance structures

Multilevel meta-analysis

Residual homogeneity test

For an overview of the statistical models behind multilevel meta-regression, see [Konstantopoulos \(2011\)](#) and [Sera et al. \(2019\)](#).

Three-level meta-regression

The model for the three-level meta-regression can be expressed as

$$\widehat{\theta}_{jkr} = \mathbf{x}_{jkr}\boldsymbol{\beta} + \mathbf{z}_{jkr}^{(3)}\mathbf{u}_j^{(3)} + \mathbf{z}_{jkr}^{(2)}\mathbf{u}_{jk}^{(2)} + \epsilon_{jkr}$$

for $j = 1, 2, \dots, M$, $k = 1, 2, \dots, m_j$, and $r = 1, 2, \dots, m_{jk}$. In this case, $\mathbf{x}_{jkr} = (1, x_{1,jkr}, \dots, x_{p-1,jkr})$ is a $1 \times p$ vector of moderators and $\boldsymbol{\beta}$ is the corresponding $p \times 1$ vector of unknown fixed-effects parameters. $\mathbf{z}_{jkr}^{(3)}$ is a $1 \times q_3$ vector of moderators associated with the level-3 $q_3 \times 1$ vector of random effects $\mathbf{u}_j^{(3)}$ (1 intercept and $q_3 - 1$ slopes), where $\mathbf{u}_j^{(3)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}^{(3)})$. Similarly, $\mathbf{z}_{jkr}^{(2)}$ is a $1 \times q_2$ vector of moderators associated with the level-2 (within-level-3) $q_2 \times 1$ vector of random effects $\mathbf{u}_{jk}^{(2)}$, where $\mathbf{u}_{jk}^{(2)} \sim N(\mathbf{0}, \boldsymbol{\Sigma}^{(2)})$. $\epsilon_{jkr} \sim N(0, \widehat{\sigma}_{jkr}^2)$ with the $\widehat{\sigma}_{jkr}^2$'s being the sampling variances.

Define the $m_{jk} \times p$ matrix $\mathbf{X}_{jk} = (\mathbf{x}'_{jk1}, \mathbf{x}'_{jk2}, \dots, \mathbf{x}'_{jkm_{jk}})'$ and the $m_{jk} \times 1$ vectors $\widehat{\boldsymbol{\theta}}_{jk} = (\widehat{\theta}_{jk1}, \widehat{\theta}_{jk2}, \dots, \widehat{\theta}_{jkm_{jk}})'$ and $\boldsymbol{\epsilon}_{jk} = (\epsilon_{jk1}, \epsilon_{jk2}, \dots, \epsilon_{jkm_{jk}})'$. The above model can now be written as

$$\widehat{\boldsymbol{\theta}}_{jk} = \mathbf{X}_{jk}\boldsymbol{\beta} + \mathbf{Z}_{jk}^{(3)}\mathbf{u}_j^{(3)} + \mathbf{Z}_{jk}^{(2)}\mathbf{u}_{jk}^{(2)} + \boldsymbol{\epsilon}_{jk}$$

where $m_{jk} \times q_3$ matrix $\mathbf{Z}_{jk}^{(3)} = (\mathbf{z}_{jk1}^{(3)'}, \mathbf{z}_{jk2}^{(3)'}, \dots, \mathbf{z}_{jkm_{jk}}^{(3)'})'$ and $m_{jk} \times q_2$ matrix $\mathbf{Z}_{jk}^{(2)} = (\mathbf{z}_{jk1}^{(2)'}, \mathbf{z}_{jk2}^{(2)'}, \dots, \mathbf{z}_{jkm_{jk}}^{(2)'})'$. The $\boldsymbol{\epsilon}_{jk}$'s have an m_{jk} -variate normal distribution with zero mean vector and a diagonal $m_{jk} \times m_{jk}$ covariance matrix $\text{Var}(\boldsymbol{\epsilon}_{jk}) = \boldsymbol{\Lambda}_{jk}$ with diagonal elements $\widehat{\sigma}_{jkr}^2$, $r = 1, 2, \dots, m_{jk}$. The covariance matrices (the $\boldsymbol{\Lambda}_{jk}$'s) are treated as known and do not require estimation. The $\boldsymbol{\Lambda}_{jk}$'s reduce to $\widehat{\sigma}_j^2$ in the case of standard meta-analysis; see [Methods and formulas of \[META\] meta summarize](#).

Let $m_j = \sum_{k=1}^{m_j} m_{jk}$ be the number of observations belonging to the j th level-3 group and define the $m_j \times p$ matrix $\mathbf{X}_j = (\mathbf{X}'_{j1}, \mathbf{X}'_{j2}, \dots, \mathbf{X}'_{jm_j})'$ and the $m_j \times 1$ vectors $\widehat{\boldsymbol{\theta}}_j = (\widehat{\theta}'_{j1}, \widehat{\theta}'_{j2}, \dots, \widehat{\theta}'_{jm_j})'$ and $\boldsymbol{\epsilon}_j = (\boldsymbol{\epsilon}'_{j1}, \boldsymbol{\epsilon}'_{j2}, \dots, \boldsymbol{\epsilon}'_{jm_j})'$ with $m_j \times m_j$ covariance matrix $\text{Var}(\boldsymbol{\epsilon}_j) = \boldsymbol{\Lambda}_j = \bigoplus_{k=1}^{m_j} \boldsymbol{\Lambda}_{jk}$, where \bigoplus is the Kronecker sum. The previous model can now be expressed as

$$\widehat{\boldsymbol{\theta}}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j^{(3)}\mathbf{u}_j^{(3)} + \mathbf{Z}_j^{(2)}\mathbf{u}_j^{(2)} + \boldsymbol{\epsilon}_j$$

where $m_j \times m_j q_2$ block-diagonal matrix $\mathbf{Z}_j^{(2)} = \bigoplus_{k=1}^{m_j} \mathbf{Z}_{jk}^{(2)}$, $m_j \times q_3$ matrix $\mathbf{Z}_j^{(3)} = (\mathbf{Z}_{j1}^{(3)'}, \mathbf{Z}_{j2}^{(3)'}, \dots, \mathbf{Z}_{jm_j}^{(3)'})'$, and $m_j q_2 \times 1$ vector of random effects at level 2 $\mathbf{u}_j^{(2)} = (\mathbf{u}_{j1}^{(2)'}, \mathbf{u}_{j2}^{(2)'}, \dots, \mathbf{u}_{jm_j}^{(2)'})' \sim N(\mathbf{0}, \mathbf{I}_{m_j} \otimes \boldsymbol{\Sigma}^{(2)})$, where \otimes is the Kronecker product.

We may eliminate the explicit reference to specific levels of hierarchy and express the previous model more compactly as

$$\widehat{\boldsymbol{\theta}}_j = \mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j + \boldsymbol{\epsilon}_j$$

where $m_j \times (q_3 + m_j q_2)$ matrix $\mathbf{Z}_j = (\mathbf{Z}_j^{(3)}, \mathbf{Z}_j^{(2)})$ and $(q_3 + m_j q_2) \times 1$ vector $\mathbf{u}_j = (\mathbf{u}_j^{(3)'}, \mathbf{u}_j^{(2)'})'$, with a $(q_3 + m_j q_2) \times (q_3 + m_j q_2)$ covariance matrix $\boldsymbol{\Sigma}_j$,

$$\boldsymbol{\Sigma}_j = \text{Var}(\mathbf{u}_j) = \begin{bmatrix} \boldsymbol{\Sigma}^{(3)} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m_j} \otimes \boldsymbol{\Sigma}^{(2)} \end{bmatrix}$$

Note that Σ_j depends on j only through its dimension. In other words, if estimates for $\Sigma^{(2)}$ and $\Sigma^{(3)}$ are available, then estimates for Σ_j , $j = 1, 2, \dots, M$ are also available.

Let $\widehat{\Sigma}_j$ be an estimate of the random-effects covariance matrix Σ_j (to be discussed later), and let $\mathbf{W}_j = (\mathbf{Z}_j \widehat{\Sigma}_j \mathbf{Z}_j' + \Lambda_j)^{-1}$. The vector of fixed-effects regression coefficients β can be estimated as

$$\widehat{\beta} = \left(\sum_{j=1}^M \mathbf{X}_j' \mathbf{W}_j \mathbf{X}_j \right)^{-1} \sum_{j=1}^M \mathbf{X}_j' \mathbf{W}_j \widehat{\theta}_j$$

The corresponding covariance matrix is given by

$$\text{Var}(\widehat{\beta}) = \left(\sum_{j=1}^M \mathbf{X}_j' \mathbf{W}_j \mathbf{X}_j \right)^{-1}$$

In the following section, we outline the estimation of the random-effects covariance matrices $\Sigma^{(2)}$ and $\Sigma^{(3)}$ (and thus of Σ_j) for the ML and REML methods.

Methods for estimating $\Sigma^{(2)}$ and $\Sigma^{(3)}$

The two estimators described below do not have a closed-form solution, and an iterative algorithm is needed to estimate $\Sigma^{(2)}$ and $\Sigma^{(3)}$.

The joint log-likelihood function of β , $\Sigma^{(2)}$, and $\Sigma^{(3)}$ for a random-effects multivariate meta-regression can be expressed as

$$\ln L_{\text{ML}}(\beta, \Sigma^{(2)}, \Sigma^{(3)}) = -\frac{1}{2} \left\{ n \ln(2\pi) + \sum_{j=1}^M \ln |\mathbf{V}_j| + \sum_{j=1}^M (\widehat{\theta}_j - \mathbf{X}_j \beta)' \mathbf{V}_j^{-1} (\widehat{\theta}_j - \mathbf{X}_j \beta) \right\}$$

where $\mathbf{V}_j = \mathbf{Z}_j \Sigma_j \mathbf{Z}_j' + \Lambda_j$, $|\mathbf{V}_j|$ is the determinant of \mathbf{V}_j , and $n = \sum_{j=1}^M \sum_{k=1}^{m_j} m_{jk}$ is the total number of observations $\widehat{\theta}_{jkr}$.

The random-effects covariance matrices $\Sigma^{(2)}$ and $\Sigma^{(3)}$ are estimated by maximizing the profile log-likelihood function obtained by treating β as known and plugging $\widehat{\beta}$ into $\ln L_{\text{ML}}(\beta, \Sigma)$ in place of β (Pinheiro and Bates [2000, chap. 2]):

$$\ln L_{\text{ML}}(\Sigma^{(2)}, \Sigma^{(3)}) = -\frac{1}{2} \left\{ n \ln(2\pi) + \sum_{j=1}^M \ln |\mathbf{V}_j| + \sum_{j=1}^M (\widehat{\theta}_j - \mathbf{X}_j \widehat{\beta})' \mathbf{V}_j^{-1} (\widehat{\theta}_j - \mathbf{X}_j \widehat{\beta}) \right\}$$

The MLE of $\Sigma^{(2)}$ and $\Sigma^{(3)}$ does not incorporate the uncertainty about the unknown regression coefficients β and thus can be negatively biased.

The REML estimator of $\Sigma^{(2)}$ and $\Sigma^{(3)}$ maximizes the restricted log-likelihood function

$$\ln L_{\text{REML}}(\Sigma^{(2)}, \Sigma^{(3)}) = \ln L_{\text{ML}}(\Sigma^{(2)}, \Sigma^{(3)}) - \frac{1}{2} \ln \left| \sum_{j=1}^M \mathbf{X}_j' \mathbf{V}_j^{-1} \mathbf{X}_j \right| + \frac{p}{2} \ln(2\pi)$$

The REML method estimates $\Sigma^{(2)}$ and $\Sigma^{(3)}$ by accounting for the uncertainty in the estimation of β , which leads to a nearly unbiased estimate of $\Sigma^{(2)}$ and $\Sigma^{(3)}$. The optimization of the above log-likelihood functions can be done using the machinery of the mixed-effects models to obtain the estimates $\hat{\beta}$, $\Sigma^{(2)}$, and $\Sigma^{(3)}$. For details, see [Pinheiro and Bates \(2000\)](#) and *Methods and formulas* of [ME] [mixed](#). When there are only two levels of hierarchy in the model and no random slopes, that is, in the context of standard meta-analysis, the above ML and REML estimators reduce to their counterparts as reported by [meta regress](#).

Random-effects covariance structures

Several covariance structures may be assumed for the $q_l \times q_l$ random-effects covariance matrix $\Sigma^{(l)}$ at a specific level of hierarchy l . The default covariance structure is [independent](#), which assumes there are q_l standard deviations to be estimated corresponding to the q_l random effects at level l . Other covariance structures are [exchangeable](#), [identity](#), [unstructured](#), and [custom matname](#). Structures that allow the random effects to be correlated ([unstructured](#), [exchangeable](#), and potentially [custom matname](#)) should be used only when adequate observations are available in order to produce stable estimates of the correlations.

For example, when there are 3 random effects at level l ($q_l = 3$), the covariance structures are

$$\begin{aligned} \text{independent } \Sigma^{(l)} &= \begin{bmatrix} \sigma_{11} & & \\ 0 & \sigma_{22} & \\ 0 & 0 & \sigma_{33} \end{bmatrix} \\ \text{exchangeable } \Sigma^{(l)} &= \begin{bmatrix} \sigma_{11} & & \\ \sigma_{21} & \sigma_{11} & \\ \sigma_{21} & \sigma_{21} & \sigma_{11} \end{bmatrix} \\ \text{identity } \Sigma^{(l)} &= \begin{bmatrix} \sigma_{11} & & \\ 0 & \sigma_{11} & \\ 0 & 0 & \sigma_{11} \end{bmatrix} \\ \text{unstructured } \Sigma^{(l)} &= \begin{bmatrix} \sigma_{11} & & \\ \sigma_{21} & \sigma_{22} & \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \end{aligned}$$

For the custom covariance structure, see [covariance\(custom matname\)](#).

Multilevel meta-analysis

The formulas presented so far are derived for the general case of multilevel meta-regression. Methods and formulas for the special case of multilevel meta-analysis (when no moderators are included) can be obtained by taking $\mathbf{x}_{jkr} = 1$, $\mathbf{z}_{jkr}^{(3)} = 1$, $\mathbf{z}_{jkr}^{(2)} = 1$, and $p = 1$. This model can be expressed as

$$\hat{\theta}_{jkr} = \beta_0 + u_j^{(3)} + u_{jk}^{(2)} + \epsilon_{jkr}$$

where $j = 1, 2, \dots, M$, $k = 1, 2, \dots, m_j$, and $r = 1, 2, \dots, m_{jk}$. When there are only two levels of hierarchy in the model, the REML and ML estimators reduce to the classical REML and ML estimators described in [META] [meta summarize](#) for constant-only models.

Residual homogeneity test

Consider a test of residual homogeneity, which mathematically translates to $H_0: \Sigma^{(l)} = \mathbf{0}_{q_l \times q_l}$, $l = 2, 3$, for the multilevel meta-regression. This test is based on the multivariate residual weighted sum of squares, Q_M , defined as

$$Q_M = \sum_{j=1}^M \left(\hat{\theta}_j - \mathbf{X}_j \hat{\beta}_f \right)' \Lambda_j^{-1} \left(\hat{\theta}_j - \mathbf{X}_j \hat{\beta}_f \right)$$

where $\hat{\beta}_f$ is a fixed-effects estimator obtained by fitting a standard fixed-effects meta-regression (see [META] **meta regress**) of the $\hat{\theta}_{jkr}$'s on the moderators defining the \mathbf{X}_j matrix.

Under the null hypothesis of residual homogeneity, Q_M follows a χ^2 distribution with $n - p$ degrees of freedom (Seber and Lee 2003, sec. 2.4). The Q_M statistic reduces to the classical residual homogeneity test statistic, Q_{res} , when there are two levels of hierarchy and no random slopes in the model (see *Residual homogeneity test in Methods and formulas* in [META] **meta regress**). It also reduces to the classical homogeneity statistic Q when no moderators are included (see *Homogeneity test in Methods and formulas* in [META] **meta summarize**).

References

- Cheung, M. W.-L. 2014. Modeling dependent effect sizes with three-level meta-analyses: A structural equation modeling approach. *Psychological Methods* 19: 211–229. <https://psycnet.apa.org/doi/10.1037/a0032968>.
- Cooper, H., J. C. Valentine, and A. Melson. 2003. The effects of modified school calendars on student achievement and on school and community attitudes. *Review of Educational Research* 73: 1–52. <https://doi.org/10.3102/00346543073001001>.
- Goldstein, H., M. Yang, R. Omar, R. M. Turner, and S. G. Thompson. 2000. Meta-analysis using multilevel models with an application to the study of class size effects. *Journal of the Royal Statistical Society, Series C* 49: 399–412. <https://doi.org/10.1111/1467-9876.00200>.
- Gutierrez, R. G., S. L. Carter, and D. M. Drukker. 2001. *sg160: On boundary-value likelihood-ratio tests*. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273. College Station, TX: Stata Press.
- Konstantopoulos, S. 2011. Fixed effects and variance components estimation in three-level meta-analysis. *Research Synthesis Methods* 2: 61–76. <https://doi.org/10.1002/jrsm.35>.
- McCurdy, M. P., W. Viechtbauer, A. M. Sklenar, A. N. Frankenstein, and E. D. Leshikar. 2020. Theories of the generation effect and the impact of generation constraint: A meta-analytic review. *Psychonomic Bulletin and Review* 27: 1139–1165. <https://doi.org/10.3758/s13423-020-01762-3>.
- Nakagawa, S., and E. S. A. Santos. 2012. Methodological issues and advances in biological meta-analysis. *Evolutionary Ecology* 26: 1253–1274. <https://doi.org/10.1007/s10682-012-9555-5>.
- Orsini, N. 2021. Weighted mixed-effects dose–response models for tables of correlated contrasts. *Stata Journal* 21: 320–347.
- Pinheiro, J. C., and D. M. Bates. 2000. *Mixed-Effects Models in S and S-PLUS*. New York: Springer.
- Raudenbush, S. W. 1984. Magnitude of teacher expectancy effects on pupil IQ as a function of the credibility of expectancy induction: A synthesis of findings from 18 experiments. *Journal of Educational Psychology* 76: 85–97. <http://doi.org/10.1037/0022-0663.76.1.85>.
- Raudenbush, S. W., and A. S. Bryk. 1985. Empirical Bayes meta-analysis. *Journal of Educational Statistics* 10: 75–98. <https://doi.org/10.2307/1164836>.
- . 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd ed. Thousand Oaks, CA: Sage.
- Seber, G. A. F., and A. J. Lee. 2003. *Linear Regression Analysis*. 2nd ed. Hoboken, NJ: Wiley.

- Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610. <https://doi.org/10.2307/2289471>.
- Sera, F., B. Armstrong, M. Blangiardo, and A. Gasparrini. 2019. An extended mixed-effects framework for meta-analysis. *Statistics in Medicine* 38: 5429–5444. <https://doi.org/10.1002/sim.8362>.
- Thompson, S. G., R. M. Turner, and D. E. Warn. 2001. Multilevel models for meta-analysis, and their application to absolute risk differences. *Statistical Methods in Medical Research* 10: 375–392. <https://doi.org/10.1177/096228020101000602>.
- Verbeke, G., and G. Molenberghs. 2000. *Linear Mixed Models for Longitudinal Data*. New York: Springer.

Also see

- [META] **meta me postestimation** — Postestimation tools for multilevel mixed-effects meta-analysis
- [META] **meta regress** — Meta-analysis regression
- [META] **meta summarize** — Summarize meta-analysis data
- [META] **meta multilevel** — Multilevel random-intercepts meta-regression
- [META] **meta** — Introduction to meta
- [META] **Glossary**
- [META] **Intro** — Introduction to meta-analysis
- [U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.

