Title

stata.com

meta multilevel — Multilevel random-intercepts meta-regression

Description

`meta multilevel` fits random-intercepts meta-analysis models, which are commonly used in practice. For fitting more complicated multilevel meta-analysis models, including random slopes, see [META] **meta meregress**. `meta multilevel` is a convenience wrapper for `meta meregress`.

`meta multilevel` 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 multilevel y, relevels(trial) essevariable(se)
```

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

```
meta multilevel y x, relevels(trial) essevariable(se)
```

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

```
meta multilevel y x, relevels(trial) esvarvariable(var)
```

Perform a three-level meta-analysis of `y` with random intercepts by `region` and by `trial` nested within `region`, and request the ML instead of the default REML estimation method

```
meta multilevel y, relevels(region trial) essevariable(se) mle
```

Perform a three-level meta-regression of `y` on `x1` and `x2` and specify a fixed standard deviation for the `trial`-within-`region` random intercepts

```
meta multilevel y x1 x2, relevels(region trial, sd(. .2))   ///
     essevariable(se)
```

Menu

Statistics > Meta-analysis

Syntax

```
meta multilevel depvar [indepvars] [if] [in], relevels(relevspec)
  { essevariable(varname) | esvarvariable(varname) } [options]
```

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

*relevels() is required. The full specification is relevels(*varlist*[, sd(# [# [...]])]).

†Either essevariable() or esvarvariable() is required.

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

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

collinear and coeflegend do not appear in the dialog box.

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

Options

Model

noconstant suppresses the constant (intercept) term from the fixed-effects model.

relevels(*varlist*[, sd(# [# [...]])]) specifies the grouping structure of the multilevel model. A random intercept corresponding to each level variable in *varlist* is included in the model. The order of *varlist* is important. The first variable is assumed to be the highest grouping level, and each subsequent variable is assumed to be nested within the previous one. For example, relevels(region study) assumes that variable region is the highest grouping level and that study is nested within region. relevels() is required.

sd(# [# [...]]) specifies fixed values for the standard deviations of the random intercepts during estimation. The order of the values # [# [...]] should correspond to the order of variables in relevels(). A missing value (.) means that the standard deviation of the corresponding random intercept is to be estimated. This suboption is useful for exploring the sensitivity of the results to different magnitudes of random-intercepts standard deviations.

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.

noretable, nofetable, noheader, and nogroup; see [ME] **mixed**.

nocnsreport; see [R] **Estimation options**.

display_options: noci, 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 multilevel 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 multilevel but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction
Examples of using meta multilevel

Introduction

For an introduction to the general multilevel meta-regression model, see *Introduction* in [META] **meta meregress**.

Let $\mathbf{x}_{jkr} = (1, x_{1,jkr}, \ldots, x_{p-1,jkr})$ be a $1 \times p$ vector of moderators and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{p-1})'$ be the corresponding $p \times 1$ vector of unknown fixed-effects regression coefficients. The three-level random-intercepts meta-regression model (Goldstein et al. [2000]; Thompson, Turner, and Warn [2001]; and Konstantopoulos [2011]) can be expressed as

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

where $j = 1, 2, \ldots, M$, $k = 1, 2, \ldots, m_j$, and $r = 1, 2, \ldots, m_{jk}$. $u_j^{(3)} \sim N(0, \tau_3^2)$, $u_{jk}^{(2)} \sim N(0, \tau_2^2)$, and $\epsilon_{jkr} \sim N(0, \widehat{\sigma}_{jkr}^2)$, with the $\widehat{\sigma}_{jkr}^2$'s being the known sampling variances (variances of the effect sizes). The random intercepts (the $u_j^{(3)}$'s, $u_{jk}^{(2)}$'s) and the sampling errors (the ϵ_{jk}'s) are independent. τ_3^2 and τ_2^2 are the random-intercepts variances at the third and second levels, respectively. Model (1) and its higher-level extensions are precisely the models that meta multilevel was designed to fit. If you wish to fit models that incorporate random slopes, see the more general command [META] **meta meregress**.

`meta multilevel` fits multilevel random-intercepts meta-regression. By default, the REML method is used to estimate the random-intercepts variances τ^2_3 and τ^2_2. Use the `mle` option to request ML estimation. REML is typically preferred over ML because it produces unbiased estimates of the random-effects variance parameters by accounting for the loss of degrees of freedom from estimating the fixed-effects vector $\boldsymbol{\beta}$.

The `relevels()` option specifies the variables that identify the different levels of hierarchy that are present in the model. For each level of hierarchy, a random intercept is added to the model. The order of the specified variables is important. The first variable is assumed to be the highest grouping level, and each subsequent variable is assumed to be nested within the previous one.

The `sd()` suboption within `relevels()` provides a flexible way to restrict specific random-intercepts standard deviations 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 suboption `sd()` in *Options*.

The sampling variances (the $\widehat{\sigma}^2_{jkr}$'s) are treated as known and do not require estimation. The variable that stores these values is specified in the `esvarvariable()` option. Alternatively, if the sampling standard errors (the $\widehat{\sigma}_{jkr}$'s) are available, then option `essevariable()` can be used instead.

For example, suppose we specify the following in Stata:

```
. meta multilevel y x1 x2, relevels(lev3var lev2var) esvarvariable(var)
```

Consider how the above specification relates to the components of (1). Variable `y` stores the values of the $\widehat{\theta}_{jkr}$'s, and the variables `x1` and `x2` represent the fixed-effects component of the model, $\mathbf{x}_{jkr}\boldsymbol{\beta}$. Three fixed-effects parameters will need to be estimated: an intercept and two coefficients corresponding to variables `x1` and `x2`, respectively. The `relevels(lev3var lev2var)` option specifies that two random intercepts are to be included in the model: one at level 3 (identified by variable `lev3var`) and another one at level 2 (identified by variable `lev2var`). These are the $u^{(3)}_j$ and $u^{(2)}_{jk}$ terms in (1). Level 1 corresponds to the participant or subject-level data, which are not available in meta-analysis. In general, if you specify L variables within `relevels()`, then $L+1$ levels of hierarchy will be present in the model, with the leftmost variable corresponding to the highest level. The `esvarvariable(var)` option specifies the variable name (`var` in our example) that stores the sampling variances (the $\widehat{\sigma}^2_{jkr}$'s) of the ϵ_{jkr}'s.

You may also use suboption `sd()` within `relevels()` to fix certain random-intercepts standard deviations at specified values during estimation, while allowing the remaining standard deviations to be freely estimated as follows:

```
. meta multilevel y x1 x2, relevels(lev3var lev2var, sd(.4 .)) esvarvariable(var)
```

Option `sd(.4 .)` specifies that the standard deviation of $u^{(3)}_j$ is to be fixed at .4 during estimation and that the standard deviation of $u^{(2)}_{jk}$ is to be estimated.

Examples of using meta multilevel

Examples are presented under the following headings:

Example 1: Three-level meta-analysis

Continuing with example 2 of [META] **meta meregress**, which explored the effect of a modified school calendar on student achievement (Cooper et al. 2003), we will fit the same model using the syntax of `meta multilevel`. Recall that this command fits models that contain random intercepts only (no random slopes). Our three-level random-intercepts model is given by

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

with $u_j^{(3)} \sim N(0, \tau_3^2)$, $u_{jk}^{(2)} \sim N(0, \tau_2^2)$, and $\epsilon_{jk} \sim N(0, \texttt{se}_{jk}^2)$. Here there is one observation (effect size) reported per school (level-2 group); therefore, $m_{jk} = 1$ in formula (1) in *Introduction*. This model can be fit using `meta multilevel` as follows:

```
. use https://www.stata-press.com/data/r18/schoolcal
(Effect of modified school calendar on student achievement)
. meta multilevel stdmdiff, relevels(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 = -51.767231  (not concave)
Iteration 2:  Log restricted-likelihood = -36.863799  (not concave)
Iteration 3:  Log restricted-likelihood = -33.383575  (not concave)
Iteration 4:  Log restricted-likelihood = -9.1062207
Iteration 5:  Log restricted-likelihood = -8.0125915
Iteration 6:  Log restricted-likelihood = -7.9587913
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
------------------------------------------------------------------------
                |      No. of       Observations per group
 Group variable |      groups    Minimum    Average    Maximum
----------------+-------------------------------------------------------
       district |          11          3        5.1         11
         school |          56          1        1.0          1
------------------------------------------------------------------------
                                                       Wald chi2(0)  =  .
Log restricted-likelihood = -7.9587239                 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
------------------------------------------------------------------------------
```

By typing `stdmdiff` after `meta multilevel`, we specified the response variable (`stdmdiff`) and the fixed-effects portion of our model, which consists of a constant term (fixed-effect intercept),

denoted by θ in (2). We could have specified `stdmdiff` *indepvars* to include additional moderators (independent variables) in the same way that we would if we were using any other estimation command. The `relevels(district school)` option defines two levels of hierarchy (the model will then have three levels, given that level 1 always corresponds to effect sizes) and includes random intercepts at both levels [the $u_j^{(3)}$ and $u_{jk}^{(2)}$ terms in (2)]. The order in which the variables are specified within `relevels()` (from left to right) is important—`meta multilevel` assumes that `school` is nested within `district`. This model was specified as follows in example 2 of [META] **meta meregress** (see that example for output interpretation):

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

In other words, the `relevels(district school)` specification in `meta multilevel` is equivalent to the `|| district: || school:` specification in `meta meregress`.

◁

▷ Example 2: Sensitivity multilevel meta-analysis

We may often wish to fix certain random-intercepts standard deviations at specified values during estimation, while allowing the remaining standard deviations to be freely estimated. This could be done as a form of sensitivity analysis to assess the impact of certain random-effects parameters on estimation overall. The `sd()` suboption within `relevels()` can be a useful tool for this. For example, we may fix the value of τ_3^2 at the value reported in example 1 (0.2550724) and check that the estimates of the other parameters match with what was reported in that example. We use options `nolog` and `noheader` to suppress the log and header output for a more compact display of the results.

```
. meta multilevel stdmdiff, relevels(district school, sd(.2550724 .))
> essevariable(se) nolog noheader
```

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: Custom	
sd(_cons)	.2550724*
school: Identity	
sd(_cons)	.1809324

(*) fixed during estimation

The order in which you specify values in `sd()` corresponds to the order in which the variables were specified within `relevels()`. In other words, the first value corresponds to the standard deviation of the random effects at the `district` level and the second value to that at the `school` level. The second `.` in `sd(.2550724 .)` means that the standard deviation of the random intercepts at the `school` level, τ_2, is free and needs to be estimated. The two outputs are essentially identical, as expected. Notice the starred note to indicate which parameter was fixed during estimation.

Next we will assess the impact of five different magnitudes in increasing order of the value of the random-effects standard deviations at the `school` level on the estimation of the other model parameters (θ and τ_3). We fit five models corresponding to fixing τ_2 at each element of matrix `val` in a loop and store their results under the names `fixsd1`, `fixsd2`, and so on.

```
. matrix val = (.01, .08, .18, .3, .6)
. forvalues i=1/5 {
  2.          quietly meta multilevel stdmdiff,
>              relevels(district school, sd(. `=val[1,`i']')) essevariable(se)
  3.          estimates store fixsd`i'
  4. }
```

We then use `estimates table` to report $\widehat{\theta}$ (option `keep(stdmdiff:_cons)`) and its standard error from the five models for ease of comparison.

```
. estimates table _all, stats(sd2) keep(stdmdiff:_cons) b(%8.3f) se(%8.3f)
```

Variable	fixsd1	fixsd2	fixsd3	fixsd4	fixsd5
_cons	0.196	0.193	0.185	0.172	0.123
	0.090	0.088	0.085	0.081	0.083
sd2	0.010	0.080	0.180	0.300	0.600

Legend: b/se

As τ_2 (`sd2` in the output) increases from 0.01 to 0.6, $\widehat{\theta}$ (`_cons` in the output) decreases from 0.196 to 0.123 and seems to be estimated with more precision (its standard error decreases). This suggests that increased variability among schools leads to a smaller overall standardized mean difference, resulting in less benefit from the modified-calendar program. Recall that a positive mean difference corresponds to higher student achievement in the group on the modified calendar.

The next table shows the estimates of $\tau_3 = \sqrt{\mathrm{Var}(u_j^{(3)})}$ for the different fixed values of τ_2. The term `lns1_1_1:_cons` (used within option `keep()`) stores the value of $\log(\tau_3)$, so we use the `eform` option to report the exponentiated value.

```
. estimates table _all, stats(sd2) keep(lns1_1_1:_cons) b(%8.3f) eform
```

Variable	fixsd1	fixsd2	fixsd3	fixsd4	fixsd5
_cons	0.288	0.278	0.255	0.215	0.000
sd2	0.010	0.080	0.180	0.300	0.600

As τ_2 (`sd2`) increases from 0.01 to 0.6, $\widehat{\tau_3}$ (`_cons`) decreases from 0.288 to nearly 0, indicating that as τ_2 increases, it will eventually capture all the variability (excluding sampling error) among the effect sizes. In this case, the district level (level 3) may be dropped from the model.

◁

Stored results

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

Scalars

`e(N)`	total number of observations
`e(k)`	number of parameters
`e(k_f)`	number of fixed-effects parameters
`e(k_r)`	number of random-effects parameters
`e(k_rs)`	number of variances
`e(ll)`	log (restricted) likelihood
`e(rank)`	rank of `e(V)`

e(ic)	number of iterations
e(sd#)	user-specified random-intercepts standard deviation (when suboption sd() of relevels() is specified)
e(df_m)	model degrees of freedom
e(chi2)	model χ^2 Wald test statistic
e(p)	p-value for model test
e(Q_M)	multilevel Cochran Q_M residual homogeneity test statistic
e(df_Q_M)	degrees of freedom for residual homogeneity test
e(p_Q_M)	p-value for residual homogeneity test
e(converged)	1 if converged, 0 otherwise

Macros

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

Matrices

e(b)	coefficient vector
e(V)	variance–covariance matrix of the estimators
e(Cns)	constraints matrix
e(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums

Functions

e(sample)	marks estimation sample

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

Matrices

r(table)	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 multilevel creates a system variable, _meta_mereg_se, that contains the sampling standard errors.

Methods and formulas

Let $\mathbf{X}_j$, $\widehat{\boldsymbol{\theta}}_j$, and $\boldsymbol{\epsilon}_j$ be defined as in *Methods and formulas* of [META] **meta meregress**. If we eliminate the explicit reference to specific levels of hierarchy, then (1) can be expressed compactly as

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

where $m_{j.} \times (m_j + 1)$ matrix $\dot{\mathbf{Z}}_j = (\mathbf{1}_{m_{j.}}, \oplus_{k=1}^{m_j}\mathbf{1}_{m_{jk}})$ and $(m_j + 1) \times 1$ vector $\dot{\mathbf{u}}_j = (u_j^{(3)}, u_{j1}^{(2)}, u_{j2}^{(2)}, \ldots, u_{jm_j}^{(2)})'$, with a $(m_j + 1) \times (m_j + 1)$ covariance matrix $\dot{\boldsymbol{\Sigma}}_j$, defined as

$$\dot{\boldsymbol{\Sigma}}_j = \mathrm{Var}\,(\dot{\mathbf{u}}_j) = \begin{bmatrix} \tau_3^2 & \mathbf{0} \\ \mathbf{0} & \tau_2^2\mathbf{I}_{m_j} \end{bmatrix}$$

The formulas used by `meta multilevel` to estimate $\boldsymbol{\beta}$, τ_3^2, and τ_2^2 are described in *Methods and formulas* of [META] **meta meregress** with $\boldsymbol{\Sigma}_j = \dot{\boldsymbol{\Sigma}}_j$, $\mathbf{Z}_j = \dot{\mathbf{Z}}_j$, and $\mathbf{u}_j = \dot{\mathbf{u}}_j$.

References

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.

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.

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.

Also see

[META] **meta me postestimation** — Postestimation tools for multilevel mixed-effects meta-analysis

[META] **meta meregress** — Multilevel mixed-effects meta-regression

[META] **meta regress** — Meta-analysis regression

[META] **meta summarize** — Summarize meta-analysis data

[META] **meta** — Introduction to meta

[META] **Glossary**

[META] **Intro** — Introduction to meta-analysis

[U] **20 Estimation and postestimation commands**