How can I estimate effect size for mixed models?

Authored by

stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I estimate effect size for mixed models?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=163606

Estimating effect size for mixed models involves determining the magnitude of the relationship between variables in a statistical model that includes both fixed and random effects. This can be achieved by calculating the standardized effect size, such as Cohen's d or Hedges' g, which measures the difference between the mean of two groups divided by the standard deviation. Other methods, such as omega squared or partial eta squared, can also be used to estimate effect size in mixed models. Additionally, conducting power analysis can provide insight into the strength of the relationship between variables in a mixed model. Overall, estimating effect size in mixed models is crucial in determining the significance and practical importance of the results obtained from these models.

How can I estimate effect size for mixed models? | Stata FAQ

Method

Note: We are unsure whether this method should be used to estimate effect sizes for predictors that vary somewhat or completely at the between-cluster level (i.e., vary at level 2), because the random intercept variance (between-cluster variance) is fixed when calculating the variance explained by the predictor. Thus, we demonstrate this method by calculating the effect size of a predictor that varies strictly at the lowest level (i.e, varies only at level 1, within-clusters/units).

This page is will show one method for estimating effects size for mixed models in Stata. Specifically, we will estimate Cohen's (f^2) effect size measure using the method described by Selya et al.(2012, see

References at the bottom) .

Here is the formula we will use to estimate the (fixed) effect size for predictor (b), (f^2_b), in a mixed model:

(R^2_{ab}) represents the proportion of variance of the outcome explained by all the predictors in a full model, including predictor (b). (1-R^2_{ab}) in the denominator thus represents the proportion of variance of the outcome not explained by the full model.

(R^2_a) represents the proportion of variance of the outcome explained by the predictors in a reduced model *with all fixed effects from the full model except for the effect of (b), and random effects constrained to be the same as those from the full model*. (R^2_{ab}-R^2_a) in the numerator is the additional proportion of variance of the outcome solely attributable to *(b)*.

Unlike linear regression models, (R^2) is not readily available from the output of mixed models, whereas residual variances typically are available, so we will calculate (R^2) from residual variances:

where (V_{null}) is the residual variance of a null model with only the intercept and random effects, and (V_{model}) is the model that includes both fixed and random effects. We can thus interpret (R^2) from a mixed model as the additional variance explained by the predictors' effects over the random effects (and intercept).

We can substitute the residual variances into the formula for (f^2_b):

We thus need the residual variances (V_{null}), (V_{ab}) and (V_{a}) to calculate our effect size (f^2_b).

The overall procedure can be summarized as:

Use `meglm` instead of `mixed`

Because of the constraint that random effects be in the reduced in null models be the same as those from the full model, we use the `meglm` command rather than `mixed`, because `meglm` allows `constraints()` whereas `mixed` does not. By default, without any further specification of `family()` or `link()`, `meglm` runs linear mixed models.

Residual variances of `meglm` models are "stored results" in Stata, so can be accessed through the `ereturn` suite of commands.

Example

For our example, we will use the hsbdemo data set.

```
use https://stats.idre.ucla.edu/stat/data/hsbdemo, clear
```

Generation of a within-classroom predictor

Predictor variables that are measured rather than assigned, like the variable read, will often vary at both the within-cluster level and at the between-cluster level. We can see the variation at both levels by running a null mixed model with read as the outcome (some output omitted throughout this page):

```
meglm read || cid:
```

```
Mixed-effects GLM Number of obs = 200
Family: Gaussian
Link: Identity
Group variable: cid Number of groups = 20

Obs per group:
min = 7
avg = 10.0
```

```
max = 12

Integration method: mvaghermite Integration pts. = 7

Wald chi2(0) = .
Log likelihood = -648.10284 Prob > chi2 = .
------------------------------------------------------------------------------
read | Coefficient Std. err. z P>|z|
-------------+----------------------------------------------------------------
_cons | 52.29178 1.994118 26.22 0.000 48.38338 56.20018
-------------+----------------------------------------------------------------
cid |
var(_cons)| 76.76532 25.14407 40.39788 145.8719
-------------+----------------------------------------------------------------
var(e.read)| 27.29169 2.876708 22.19774 33.55461
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 201.39 Prob >= chibar2 = 0.0000
```

We can see that there is significant variation in read both within-classroom, var(Residual) = 27.29169, and between-classrooms, var(_cons) = 76.76589. To demonstrate how to calculate this effect size measure, we want to use the component of read that varies only within-classroom.

To create such a variable, we need to create a classroom-centered (i.e., group-centered or cluster-centered) version of read, by subtracting the classroom means of read from the observed value of read.

```
* generate classroom means for read
egen readmean = mean(read), by(cid)
* generate classroom-centered read_within, which is a within-classroom variable
gen read_within = read - readmean
```

We can see that the within-classroom predictor read_within contains no between-classroom variation by running a null mixed model with read_within as the outcome:

```
meglm read_within || cid:
```

```
Mixed-effects GLM Number of obs = 200
Family: Gaussian
Link: Identity
Group variable: cid Number of groups = 20
```

```
Obs per group:
min = 7
avg = 10.0
max = 12

Integration method: mvaghermite Integration pts. = 7

Wald chi2(0) = .
Log likelihood = -603.91288 Prob > chi2 = .
--------------------------------------------------------------------------------
----
read_within | Coefficient Std. err. z P>|z|
-------------------+------------------------------------------------------------
-----
_cons | 1.34e-07 .3504516 0.00 1.000 -.6868723 .6868726
-------------------+------------------------------------------------------------
-----
cid |
var(_cons)| 1.15e-16 .000704 . .
-------------------+------------------------------------------------------------
-----
var(e.read_within)| 24.56326 2.456326 20.19137 29.88176
--------------------------------------------------------------------------------
----
```

LR test vs. linear model: chibar2(01) = 0.00 Prob >= chibar2 = 1.0000

We see that var(_cons)=0, which means that there is no between-classroom variation in this variable, but there is still within-classroom variation, as var(Residual)=24.56326.

Estimating the effect size of read_within

Full model

Following the procedure outlined at the top of this page, we first run the full model with both read_within and female as fixed effects, and random intercepts by cid:

meglmwrite read_within female || cid:

Mixed-effects GLM Number of obs = 200
Family: Gaussian
Link: Identity
Group variable: cid Number of groups = 20

Obs per group:
min = 7
avg = 10.0

```
max = 12

Integration method: mvaghermite Integration pts. = 7

Wald chi2(2) = 51.66
Log likelihood = -623.13939 Prob > chi2 = 0.0000
----------------------------------------------------------------------
write | Coefficient Std. err. z P>|z|
------------+----------------------------------------------------------
read_within | -.2633729 .0662426 -3.98 0.000 -.393206 -.1335398
female | 3.815357 .6956793 5.48 0.000 2.45185 5.178863
_cons | 50.82172 1.800351 28.23 0.000 47.2931 54.35034
------------+----------------------------------------------------------
cid |
var(_cons)| 59.73553 19.56032 31.44184 113.49
------------+----------------------------------------------------------
var(e.write)| 21.26528 2.241403 17.29629 26.14505
----------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 202.49 Prob >= chibar2 = 0.0000
```

In the output above, we see that the residual variance, var(Residual), is the fifth coefficient. Coefficients are typically stored in matrix e(b). We store these results in our own matrix ab, which we then view with `matrix list`:

```
matrix ab=e(b)
matrix list ab
```

```
ab
write: write: write: /: /:
var(
read_within female _cons _cons) var(e.write)
y1 -.26337293 3.8153565 50.821722 59.735531 21.265284
```

We capture the residual error variance of the full model in global macro Vab:

```
global Vab = ab
```

We also need to capture the random intercept variance, because in this method, the reduced model is constrained to have the same random effects as the full model, so that the only effect that differs between the two models is the predictor that has been removed (whose effect size we are estimating). We see in the output table and the matrix listing for e(b) that the random intercept variance is fourth coefficient. Here, we set up a `constraint`, labeled constraint 1, that will fix the random intercept variance in the reduced to be equal to this random intercept variance. We will use this constraint for the reduced and null models:

constraint 1 _b)]= ab

Note: To get the name of the random intercept variance coefficient to use in `constraint`, run the `meglm` model with the option `coeflegend`:

meglm write read_within female || cid:, coeflegend

Reduced model with constrained random intercept variance

Next we run a model without the effect of interest, female, but with random intercept variance constrained (using constraint 1 defined above) to be the same as the full model above.

meglm write female || cid:, constraints(1)

Mixed-effects GLM Number of obs = 200
Family: Gaussian
Link: Identity

```
Group variable: cid Number of groups = 20

Obs per group:
min = 7
avg = 10.0
max = 12

Integration method: mvaghermite Integration pts. = 7

Wald chi2(1) = 33.01
Log likelihood = -630.7138 Prob > chi2 = 0.0000
( 1) var(_cons) = 59.73553
-------------------------------------------------------------------------------------
write | Coefficient Std. err. z P>|z|
-------------+----------------------------------------------------------------------
female | 4.139538 .7205262 5.75 0.000 2.727333 5.551744
_cons | 50.64274 1.805927 28.04 0.000 47.10319
54.18229
-------------+----------------------------------------------------------------------
cid |
var(_cons)| 59.73553 (constrained)
-------------+----------------------------------------------------------------------
var(e.write)| 23.13675 2.438413 18.81885 28.44537
-------------------------------------------------------------------------------------
```

Notice how the random intercept variance, var(_cons) has been constrained to be the same as the full model above.

In this case the residual variance is the *fourth* coefficient (since we no longer have a coefficient for read_within).

```
matrix a = e(b)
matrix li a

a
write: write: /: /:
var(
female _cons _cons) var(e.write)
y1 4.1395383 50.642738 59.73553 23.136748
```

We will capture the residual variance in global macro Va

```
global Va = a
```

Null model

Finally, we remove all predictors from the model and retain only the random intercepts. We still constrain the variance of the random intercepts to be the same as the

full model:

```
meglm write || cid:, constraints(1)

Mixed-effects GLM Number of obs = 200
Family: Gaussian
Link: Identity
Group variable: cid Number of groups = 20

Obs per group:
min = 7
avg = 10.0
max = 12

Integration method: mvaghermite Integration pts. = 7

Wald chi2(0) = .
Log likelihood = -645.91466 Prob > chi2 = .
( 1) var(_cons) = 59.73553
------------------------------------------------------------------------------
write | Coefficient Std. err. z P>|z|
-------------+----------------------------------------------------------------
_cons | 52.92088 1.767778 29.94 0.000 49.4561 56.38566
-------------+----------------------------------------------------------------
cid |
```

```
var(_cons)| 59.73553 (constrained)
-------------+----------------------------------------------------------------
var(e.write)| 27.29963 2.877335 22.20453 33.56387
------------------------------------------------------------------------------
```

Now the residual variance is the third coefficient:

```
matrix null=e(b)
matrix list null
```

```
null
write: /: /:
var(
_cons _cons) var(e.write)
y1 52.920879 59.73553 27.299633
```

We capture the residual variance of the null model in global macro Vnull

```
global Vnull = null
```

Calculation of effect size and (R^2) values

We now have the residual variances, (V_{ab}), (V_{a}), and (V_{null}), necessary to calculate the effect size of predictor read_within, (f^2_b).

Because they are interesting quantities themselves, we first calculate (R^2_{ab}) and (R^2_a) and display their values.

```
global R2ab = ($Vnull - $Vab)/$Vnull
global R2a = ($Vnull - $Va)/$Vnull
display "Proportion explained full model = $R2ab"

Proportion explained full model = .2210414092672042

display "Proportion explained reduced model = $R2a"

Proportion explained reduced model = .1524886778022447
```

Finally, we compute the effect size and display its value:

```
global f2b = ($R2ab - $R2a)/(1-$R2ab)
display "Effect size = $f2b"

Effect size = .0880056170899526
```

Reference

Selya AS, Rose JS, Dierker LC, Hedeker D, Mermelstein RJ. A Practical Guide to Calculating Cohen's f2, a Measure of Local Effect Size, from PROC MIXED. Frontiers in Psychology 2012.