Multilevel and Mixed Models in Stata

Bill Rising

StataCorp LP

2013 Japanese Stata Users Group Meeting
Tokyo

Goals

- Learn about mixed models
  - We will work primarily with linear models
  - The syntax for generalized linear models is very similar
- Start simple, build up more complex models
  - Random intercept
  - Random intercept and slope
  - Multilevel models
  - Crossed effects
- Work with different error and random-effect covariance structures along the way

What are Linear Mixed Models?

- Linear mixed models are a generalization of linear models
- A standard linear model looks like

$$y_i = \beta_0 + \beta_1 x_{i1} \dots \beta_p x_{ip} + \varepsilon_i$$

  where $1 \leq i \leq n$, the ε_i are normally distributed and uncorrelated with each other (or the x_{ij})
  - The β's are considered to be fixed unknowns which must be estimated together with σ^2
- In a mixed model, there are multiple groups (or panels or individuals) and one or more β can vary across whatever grouping is present
  - We see one particular realization of the random β's in any one dataset

Upshot of a Mixed Model

- Mixed models are called “mixed”, because the β’s are a mix of fixed parameters and random variables
  - The terms “fixed” and “random” are being used in the statistics-biostatistics sense:
  - A fixed coefficient is an unknown constant of nature
  - A random coefficient is one which varies from sample of groups to sample of groups
- The models can have some added complexity
  - Correlations between different random β’s
  - Multiple levels of nesting withing the groups
- The random β are not estimated, though they can be predicted
- We will run through some examples to show how these work in Stata

Introductory Example

- Open up the `jsp2` dataset downloaded earlier

```
. use jsp2
```

  - These are the London Education Authority Junior School Project data as described in Mortimore *et al*.

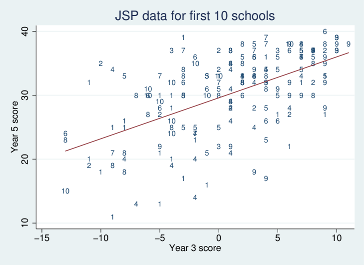
- A quick `codebook` command shows there are 48 different schools and 887 different students with no missing values

```
. codebook
```

A Partial Graph

- Here is what the data look like for the first 10 schools

```
. do schoolgph
```



- More complex commands will be put in do-files to save typing

Linear Regression

- If all we would like is to predict 5th-year math scores from 3rd-year scores, we can run a simple linear regression
- We should at least acknowledge that we have groups, as this affects independence of errors across students

```
. regress math5 math3, vce(cluster school)
```

Thinking about Better Models

- If we believe that there are differences from school to school, we should include this in the model
- Here is a start:
  $\texttt{math5}_{ij} = \beta_0 + \beta_1\texttt{math3}_{ij} + u_i + \epsilon_{ij}$
  - Here, the i represents the school and j being the pupil within each school
- This model assumes schools add a random offset to 5th-year scores
- This is called a random-intercept model, because the intercept is different from school to school, but the slope of the regression line for each school is fixed at β_1
  - We could think of β_0 as random, or, as we have above, think of an overall constant with a random offset for each school

The `mixed` Command

- The `mixed` command is made for linear mixed models
- Here the general syntax for a single-level model such as this:

  `mixed` *fixed* `||` *grpvar*: *random* , *options*

  - The random coefficients are implied by the *grpvar*
  - The coefficients in the *random* portion are the random effects
  - The options control variance structures and estimation methods, for the most part

Fitting the Random Intercept Model

- For our model, math5 and math3 are fixed
- The constant coefficient is allowed to vary across schools
- So, the simplest syntax for the model is

```
. mixed math5 math3 || school:
```

- Notice that because our random coefficient model is the constant model, there are no terms specified
    - The constant model is the same as intercept-only model
- We want to keep these results off to the side for a later test

```
. estimates store randint
```

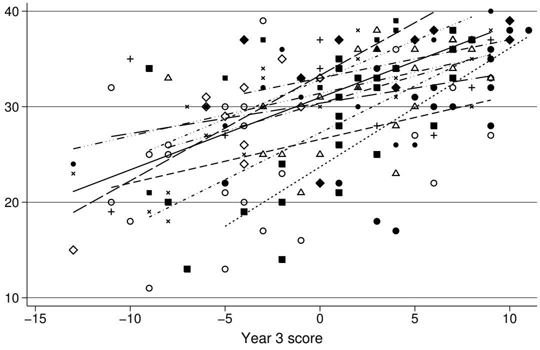
Extending the Model: Concept

- A random-intercept model is a bit of an oversimplification
  - On average, the `math5` school is different by the same amount for all students at two different schools whose `math3` scores match
  - This is regardless of the particular `math3` itself

Extending the Model: Picture

- We can overlay a series of lines to see that slopes look different from school to school

```
. do schslope
```



Extending the Model: Formal Notation

- If teaching were different from school to school, it would make sense to have different slopes for each of the schools, also
- Our formulation would then be

$$\texttt{math5}_{ij} = \underbrace{\beta_0 + \beta_1 \texttt{math3}_{ij}}_{\text{fixed}} + \underbrace{u_{0i} + u_{1i}\texttt{math3}_{ij}}_{\text{random}} + \epsilon_{ij}$$

- We will be able to test whether this extra complexity is needed

Fitting a Random Slope Model

- Telling Stata to fit this model is simple enough
  - We just need to say that the random portion has a slope for `math3`
- Here is the command

```
. mixed math5 math3 || school: math3
```

- We will want these results put aside also

```
. estimates store randslope
```

Was the Added Complication Worthwhile?

- We can run a likelihood ratio test to see if adding the extra parameter was worthwhile
- This is done with the `lrtest` command:

```
. lrtest randint randslope
```

- The random-slope model is more worthwhile

A Somewhat Unrealistic Situation

- We have two random effects, now u_{i0} and u_{i1}
- By default, `mixed` assumes that they are independent
- This is always unrealistic whenever the dependent variable is not mean-centered
  - Small slopes will be associated with large (high) intercepts
  - Large slopes will be associsated with small (low) intercepts
- It can also be unrealistic even if the dependent variable is mean-centered

Possible Covariance Structures

- For us to state a dependence, we use the `cov` option
- `independent`: (default) Each random effect has its own variance; all are independent
  - Typical between multiple random effects
- `identity`: All random effects share the same single variance; all are independent
  - We'll see the use for this below
- `exchangeable`: All random effects share a single variance. All share the same covariance with each other
  - Useful for nested intercept-only models
- `unstructured`: All variances and covariances may be different
  - Typical for slope models

Fitting with Covariances

- Let's use the unstructured covariance

```
. mixed math5 math3 || school: math3, cov(uns)
```

- Notice now that there can be a correlation between the slopes and the intercepts
- Let's store the estimates

```
. estimates store randslopex
```

- Note: In general, when fitting random slope models, it makes more sense to use the unstructured covariance than the indepedent variance

Picking a Model

- Once again, we can test whether the added correlation estimate was worthwhile
  - From the confidence interval, it appears that this is not the case
- The command is similar to before

```
. lrtest randslope randslopex
```

- As expected, it was worthwhile to include the correlation term

Prediction

- After fitting a model, we might want to get fitted values, residuals, and the like
- For mixed effects models, there is more to this than meets the eye
  - There needs to be a way to split fixed and random effects
- The prediction is still done with `predict`, but there is now more to think about

Options for predict

- Here is our model, again

$$\texttt{math5}_{ij} = \underbrace{\beta_0 + \beta_1 \texttt{math3}_{ij}}_{\text{fixed}} + \underbrace{u_{0i} + u_{1i}\texttt{math3}_{ij}}_{\text{random}} + \epsilon_{ij}$$

- Here are the options
  - xb predicts the fixed effects: $\widehat{\beta}_0 + \widehat{\beta}_1 \texttt{math3}_{ij}$
  - reffects estimates the random effects $\widehat{u}_{0i}$ and $\widehat{u}_{1i}$
    - We need to specify 2 variables for our model (or use a wildcard)
    - These predict the empirical bayes estimates/BLUPs
  - reses estimates the standard errors for the random effects
  - fitted estimates $\widehat{\beta}_0 + \widehat{\beta}_1 \texttt{math3}_{ij} + \widehat{u}_{0i} + \widehat{u}_{1i}\texttt{math3}_{ij}$
  - residuals estimates $\widehat{\epsilon}_{ij}$

Showing the Effects of Random Effects, Preparation

- We can predict the fitted values

```
. predict scorehat, fitted
```

- Then sort

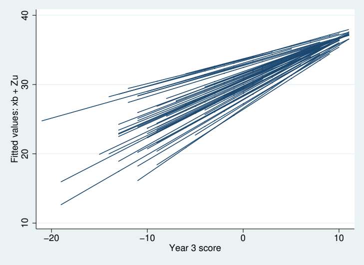
```
. sort school math3
```

- Sorting ensures the observations are ascending within school

Showing the Effects of Random Effects, Graphing

- Here is a graph which gives the idea of the effects

```
. twoway line scorehat math3, connect(ascending)
```



More Complex Models

- Now we would like to look at two different types of mixed models
  - Models with nested effects
  - Models with crossed effects
- We would also like to use more complex models for the error terms

Setup for a Three-level Model

- Open up the following dataset

```
. use productivity, clear
```

- Take a look at what it contains

```
. codebook
```

- These are gross state products measured from 1970–1986, with the 48 continental states nested within 9 regions
- We would like to fit GSP as a function of some of the covariates, making sure that we nest the states within regions
- Note: Most of the measurements are stored as logarithms, because we would really like to fit a multiplictative model

Aside: Cobb-Douglas Production Function

- Suppose we would like to model production P as a function
- The Cobb-Douglas Production is a multiplicative model:

$$P = AK^{\beta_1}L^{\beta_2}\exp(\varepsilon)$$

  - In this formulation, K is capital and L is labor resources
- Taking logs turns this into a linear model

$$\ln(P) = \ln(A) + \beta_1 \ln(K) + \beta_2 \ln(L) + \varepsilon$$

- Here we'll mimic the model Baltagi *et al.* (2001) used for Gross State Products (GSPs)
  - We will be using logs to get a multiplicative model, split the capital resources into multiple classifications

Dataset and Model

- We would like to fit the following model for the GSPs

$$\texttt{gsp}_{ijk} = \beta_0 + \beta_1\texttt{private}_{ijk} + \cdots + \beta_6\texttt{unemp}_{ijk} + u_i + v_{j(i)} + \varepsilon_{ijk}$$

  for $k = 1, 2, \ldots, 17$ annual measurements on $j = 1, \ldots, M_i$ states nested within $i = 1, \ldots, 9$ regions

- So we have 2 levels of random intercepts: one due to the regions, and another due to the state within region

Fitting the Model in Stata

- We fit this as before with fixed and random portions, but we nest the random portions from highest level down:

```
. mixed gsp private emp hwy water other unemp ///
     || region: || state:
```

- There is no real added complication to fitting this model
    - We might want to treat the years as random effects, but they are crossed with the regions
    - This brings us to the next topic—so we will come back to this below

Crossed Random Effects—Intro

- Here is the situation and model—a common one in the econ literature
- Grunfeld (1958) analyzed data on 10 large U.S. corporations collected annually from 1935 to 1954 to investigate how investment (I) depends on market value (M) and capital stock (C)
- For this model, we would like random effects (intercepts) due to firm and year. However, we want the year effect to be the same across all firms, not nested within firms
- This leads to the following model

$$I_{ij} = \beta_0 + \beta_1 M_{ij} + \beta_2 C_{ij} + u_i + v_j + \varepsilon_{ij}$$

for $i = 1, \ldots, 10$ firms measured over $j = 1, \ldots, 20$ years

Open Up the Dataset

- Open up the dataset

```
. use grunfeld
```

- Take a look at it

```
. codebook
```

- Our variables have names which are more descriptive than the usual one-letter abbreviations:
  - `invest` for investment (I)
  - `mvalue` for market value (M)
  - `cstock` for capital stock (C)

A New Specification

- Crossed effects mean there are no independent panels
  - This is done by specifying `_all` as the panel variable
- This means that group-specific random effects need to be treated as random coeffcients on indicator variables identifying each group
  - This is done by using `R.`*variable*

Leading Into the Crossed Model

- Start by fitting a model which has only companies as the random effects

```
. mixed invest mvalue cstock || company:
```

- This is the same as taking the dataset as one big panel and treating each company as nested within the superpanel and forcing all the random effects to have the same variance

```
. mixed invest mvalue cstock || _all: R.company
```

- This now points towards how to fit a crossed model:
  - Each higher level is nested within the previous level, so
    - Start with `|| _all: R.`*something*
    - Repeat until the last term
    - The last term can be `||` *lasthing*`:`

Fitting the Crossed Model

- Life is easy here, because we have just two crossed terms
- We can specify the model with 2 random intercepts following the above schema

```
. mixed invest mvalue cstock ///
    || _all: R.company || year:
```

- This is not only a nice touch, it saves a lot of computations over the full specification

```
. mixed invest mvalue cstock ///
    || _all: R.company || _all: R.year
```

- To save the most effort, put the factor with the most levels second (as we did)

Back to the State GNP model

- We can now fit the productivity model without ignoring the years entirely
- Open up the productivity dataset again

```
. use productivity
```

- Fit the model crossing years with regions, and nesting states within regions

```
. mixed gsp private emp hwy water other unemp ///
      || _all: R.year || region: || state:
```

Playing with Residuals

- We will use exercise 3.5 from Rabe-Hesketh and Skrondal (2008), which mimics a study by Dempster (1984) to look at the effect of a drug on birthweights of rat pups
- The weights of the pups depend on
  - Litter size and dosage (both at the litter level)
  - Sex of the pup (at the pup level)
- Here are the data

```
. use rats
```

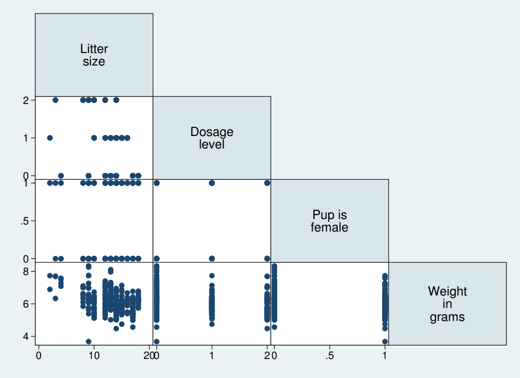
- Take a look

```
. codebook
```

A Picture of the Data

- Here is a start of looking at the data

```
. graph matrix size dose female weight, half
```



An Initial Model

- We can start by fitting a standard random-intercept model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1\texttt{dose1}_{ij} + \beta_2\texttt{dose2}_{ij} + \beta_3\texttt{size}_{ij} + \beta_4\texttt{female}_{ij} + u_i + \varepsilon_{ij}$$

  for $i = 1, \ldots, 27$ litters and $j = 1, \ldots, n_j$ pups within each litter
- Here is the model

```
. mixed weight i.dose size female || litter: , base
```

Looking at Residuals

- We should peek at the residuals to see if there are any problems
- Get the fixed portion

```
. predict fixed, xb
```

- Get the residuals

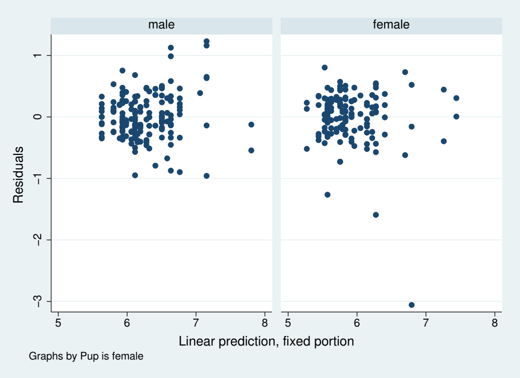
```
. predict resid, residuals
```

- Make a graph (next slide)

The Residual vs. Fitted Plot

- Here is the graph

```
. twoway scatter resid fixed, by(female)
```



- We might want to allow the variance of the residuals to vary by

Allowing for Differing Variances

- We would like to have different variances by sex
- This is simple enough:
  - Tell Stata a structure for the residuals
    - Here it makes sense that the structure is independent
    - The full list of possible choices are `independent` (the default), `exchangeable`, `ar #`, `ma #`, `unstructured`, `exponential`, `banded #`, and `toeplitz #`
    - Use a by option within the variance structure to allow differing variances
- Here is the model

```
. mixed weight i.dose size female || litter: , ///
    residuals(independent, by(female)) base
```

Non-Linear Models

- For general mixed effect effects models
  - `mixed` fits linear models
  - For binary data, Stata has the `melogit` command
  - For count data, Stata has the `mepoisson` command
  - More generally, Stata hs the `meglm` command for fitting generalized linear mixed effect models
    - This is new in Stata 13
- These all have pretty much the same syntax
  - The only differences are those which would be seen in the differences between `regress`, `logit` or `logistic`, and `poisson`

One Example for a Binary Response

- We will mimic the analysis by Ng *et al.* (2006) of the 1989 Bangledesh fertility survey
- Here are the data

```
. use bangladesh, clear
```

- Take a look

```
. codebook
```

- Data on contraception use as collected in 60 districts containing both urban and rural areas

A Binary Model Implementation

- For woman j in district i, consider this model for $\pi_{ij} := \text{Pr}[\text{cuse}_{ij} = 1]$:

$$\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \texttt{urban}_{ij} + \beta_2 \texttt{age}_{ij} + \beta_3 \texttt{child1}_{ij} + \beta_4 \texttt{child2}_{ij} + \beta_5 \texttt{child3}_{ij} + u_i + v_i \texttt{urban}_{ij}$$

  - The u_i represent 60 district-specific random effects
  - The v_i represent 60 district-specific effects of being from an urban area In other words, for rural areas the "district effect" is u_i; for urban areas it is $u_i + v_i$
- We can use `melogit` to fit this model

```
. melogit c_use age urban child* ///
    || district: urban, cov(unstructured) or
```

Aside: Numerical Integration

- Evaluating the log likelihood requires integrating out the random effects
- The numerical integration has been sped up in Stata 13
- Still, there are multiple intgration techniques which can be used
  - Mean-variance adaptive Gauss-Hermite quadrature
    - The default unless fitting a crossed random-effects model
  - Mode-curvature adaptive Gauss-Hermite quadrature,
  - Nonadaptive Gauss-Hermite quadrature, and
  - The Laplacian approximation (default for crossed random-effects models).

Conclusion

- `mixed` fits linear mixed models
- `melogit` and `mepoisson` fit two types of non-linear models
- Other types of non-linear models are available including `meglm` for multilevel generalized linear models
- These can all both nested and crossed models
- Error terms can be modelled
- `predict` is used to get predictions and residuals after fitting a model