Linear Mixed Models in Stata

Roberto G. Gutierrez

Director of Statistics
StataCorp LP

Fourth German Stata Users Group Meeting

Outline

1. The Linear Mixed Model
2. One-Level Models
3. Two-Level Models
4. Factor Notation
5. A Glimpse at the Future

Outline

Outline

1. The Linear Mixed Model
2. One-Level Models
3. Two-Level Models
4. Factor Notation
5. A Glimpse at the Future

Outline

Outline

Definition

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \epsilon$$

where

$\mathbf{y}$ is the $n \times 1$ vector of responses

$\mathbf{X}$ is the $n \times p$ fixed-effects design matrix

β are the fixed effects

$\mathbf{Z}$ is the $n \times q$ random-effects design matrix

$\mathbf{u}$ are the random effects

ϵ is the $n \times 1$ vector of errors such that

$$\begin{bmatrix} \mathbf{u} \\ \epsilon \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \sigma_\epsilon^2 \mathbf{I}_n \end{bmatrix}\right)$$

Variance components

- Random effects are not directly estimated, but instead characterized by the elements of **G**, known as *variance components*
- You can, however "predict" random effects
- As such, you fit a mixed model by estimating β, σ_ϵ^2, and the variance components in **G**

Variance components

- Random effects are not directly estimated, but instead characterized by the elements of **G**, known as *variance components*

- You can, however "predict" random effects

- As such, you fit a mixed model by estimating β, σ_ϵ^2, and the variance components in **G**

Variance components

- Random effects are not directly estimated, but instead characterized by the elements of **G**, known as *variance components*
- You can, however "predict" random effects
- As such, you fit a mixed model by estimating $\boldsymbol{\beta}$, σ_ϵ^2, and the variance components in **G**

Panel representation

- Classical representation has roots in the design literature, but can make model specification hard

- When the data can be thought of as M independent panels, it is more convenient to express the mixed model as (for $i = 1, ..., M$)

$$\mathbf{y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{u}_i + \epsilon_i$$

where $\mathbf{u}_i \sim N(\mathbf{0}, \mathbf{S})$, for $q \times q$ variance $\mathbf{S}$, and

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \mathbf{S}$$

Panel representation

- Classical representation has roots in the design literature, but can make model specification hard
- When the data can be thought of as M independent panels, it is more convenient to express the mixed model as (for $i = 1, \ldots, M$)

$$\mathbf{y}_i = \mathbf{X}_i\beta + \mathbf{Z}_i\mathbf{u}_i + \epsilon_i$$

where $\mathbf{u}_i \sim N(\mathbf{0}, \mathbf{S})$, for $q \times q$ variance $\mathbf{S}$, and

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Z}_M \end{bmatrix}; \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \end{bmatrix}; \quad \mathbf{G} = \mathbf{I}_M \otimes \mathbf{S}$$

Using the panel representation

- For example, take a random intercept model. In the classical framework, the random intercepts are random coefficients on indicator variables identifying each panel

- It is better to just think at the panel level and consider M realizations of a random intercept

- This generalizes to more than one level of nested panels

- Issue of terminology for multi-level models

Using the panel representation

- For example, take a random intercept model. In the classical framework, the random intercepts are random coefficients on indicator variables identifying each panel
- It is better to just think at the panel level and consider M realizations of a random intercept
- This generalizes to more than one level of nested panels
- Issue of terminology for multi-level models

Using the panel representation

- For example, take a random intercept model. In the classical framework, the random intercepts are random coefficients on indicator variables identifying each panel
- It is better to just think at the panel level and consider M realizations of a random intercept
- This generalizes to more than one level of nested panels
- Issue of terminology for multi-level models

Using the panel representation

- For example, take a random intercept model. In the classical framework, the random intercepts are random coefficients on indicator variables identifying each panel
- It is better to just think at the panel level and consider M realizations of a random intercept
- This generalizes to more than one level of nested panels
- Issue of terminology for multi-level models

One-level models

Example

- Consider the Junior School Project data which compares math scores of various schools in the third and fifth years
- Data on $n = 887$ pupils in $M = 48$ schools

Let's fit the model

$$\texttt{math5}_{ij} = \beta_0 + \beta_1 \texttt{math3}_{ij} + u_i + \epsilon_{ij}$$

for $i = 1, ..., 48$ schools and $j = 1, ..., n_i$ pupils. u_i is a random effect (intercept) at the school level

One-level models

Example

- Consider the Junior School Project data which compares math scores of various schools in the third and fifth years
- Data on $n = 887$ pupils in $M = 48$ schools

Let's fit the model

$$\texttt{math5}_{ij} = \beta_0 + \beta_1 \texttt{math3}_{ij} + u_i + \epsilon_{ij}$$

for $i = 1, ..., 48$ schools and $j = 1, ..., n_i$ pupils. u_i is a random effect (intercept) at the school level

```
. xtmixed math5 math3 || school:
Performing EM optimization:
Performing gradient-based optimization:
Iteration 0:   log restricted-likelihood = -2770.5233
Iteration 1:   log restricted-likelihood = -2770.5233
Computing standard errors:
Mixed-effects REML regression                   Number of obs      =       887
Group variable: school                          Number of groups   =        48
                                                Obs per group: min =         5
                                                               avg =      18.5
                                                               max =        62
                                                Wald chi2(1)       =    347.21
Log restricted-likelihood = -2770.5233          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6088557   .0326751    18.63   0.000     .5448137    .6728978
       _cons |   30.36506   .3531615    85.98   0.000     29.67287    31.05724
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
school: Identity             |
                   sd(_cons) |   2.038896   .3017985      1.525456    2.72515
-----------------------------+------------------------------------------------
                sd(Residual) |   5.306476   .1295751      5.058495   5.566614
------------------------------------------------------------------------------
LR test vs. linear regression: chibar2(01) =    57.59 Prob >= chibar2 = 0.0000
```

Adding a random slope

- For the most part, the previous is what you would get using `xtreg`

- Consider instead the model

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

- In essence, each school has its own random regression line such that the intercept is $N(\beta_0, \sigma_0^2)$ and the slope on `math3` is $N(\beta_1, \sigma_1^2)$

Adding a random slope

- For the most part, the previous is what you would get using `xtreg`

- Consider instead the model

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

- In essence, each school has its own random regression line such that the intercept is $N(\beta_0, \sigma_0^2)$ and the slope on `math3` is $N(\beta_1, \sigma_1^2)$

Adding a random slope

- For the most part, the previous is what you would get using `xtreg`

- Consider instead the model

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

- In essence, each school has its own random regression line such that the intercept is $N(\beta_0, \sigma_0^2)$ and the slope on `math3` is $N(\beta_1, \sigma_1^2)$

```
. xtmixed math5 math3 || school: math3
```

(*output omitted*)

```
Mixed-effects REML regression                 Number of obs      =       887
Group variable: school                        Number of groups   =        48

                                              Obs per group: min =         5
                                                             avg =      18.5
                                                             max =        62

                                              Wald chi2(1)       =    192.62
Log restricted-likelihood = -2766.6442        Prob > chi2        =    0.0000
```

math5	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
math3	.6135888	.0442106	13.88	0.000	.5269377	.7002399
_cons	30.36542	.3596906	84.42	0.000	29.66044	31.0704

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
school: Independent				
sd(math3)	.1911842	.0509905	.113352	.3224593
sd(_cons)	2.073863	.3078237	1.550372	2.774112
sd(Residual)	5.203947	.1309477	4.953521	5.467034

```
LR test vs. linear regression:       chi2(2) =    65.35   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference
```

Predict

Random effects are not estimated, but the can be predicted (BLUPs)

```
. predict r1 r0, reffects
. describe r*
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
r1              float  %9.0g                  BLUP r.e. for school: math3
r0              float  %9.0g                  BLUP r.e. for school: _cons
. gen b0 = _b[_cons] + r0
. gen b1 = _b[math3] + r1
. bysort school: gen tolist = _n==1
```

Predict

Random effects are not estimated, but the can be predicted (BLUPs)

```
. predict r1 r0, reffects

. describe r*

              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
r1              float  %9.0g                  BLUP r.e. for school: math3
r0              float  %9.0g                  BLUP r.e. for school: _cons

. gen b0 = _b[_cons] + r0

. gen b1 = _b[math3] + r1

. bysort school: gen tolist = _n==1
```

Predict

```
. list school b0 b1 if school<=10 & tolist
```

	school	b0	b1
1.	1	27.52259	.5527437
26.	2	30.35573	.5036528
36.	3	31.49648	.5962557
44.	4	28.08686	.7505417
68.	5	30.29471	.5983001
93.	6	31.04652	.5532793
106.	7	31.93729	.6756551
116.	8	30.83009	.6885387
142.	9	27.90685	.6950143
163.	10	31.31212	.7024184

Option fitted

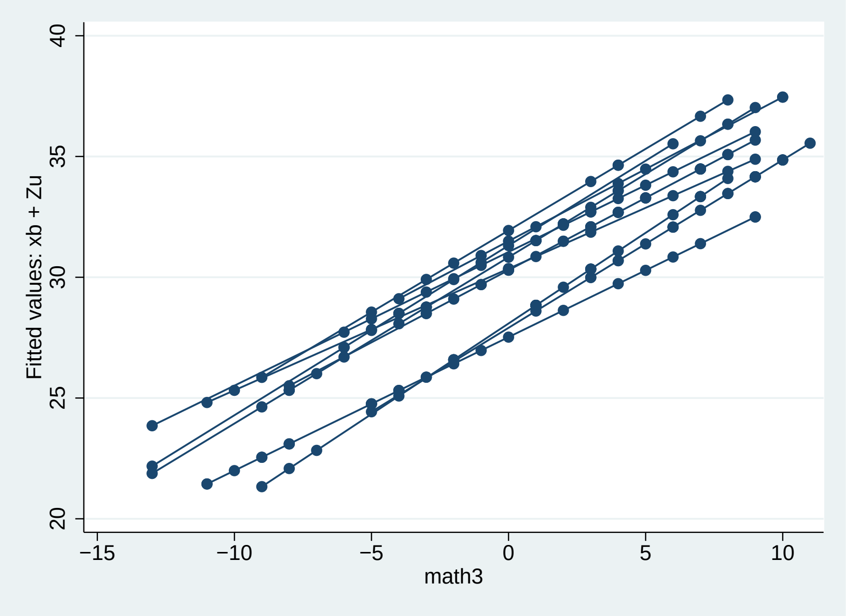
We could use these intercepts and slopes to plot the estimated lines for each school. Equivalently, we could just plot the "fitted" values

```
. predict math5hat, fitted
. sort school math3
. twoway connected math5hat math3 if school<=10, connect(L)
```

Option fitted

We could use these intercepts and slopes to plot the estimated lines for each school. Equivalently, we could just plot the "fitted" values

```
. predict math5hat, fitted
. sort school math3
. twoway connected math5hat math3 if school<=10, connect(L)
```


40
35
30
25
20
Fitted values: xb + Zu
−15
−10
−5
0
5
10
math3

Covariance structures

In our previous model, it was assumed that u_{0i} and u_{1i} are independent. That is,

$$\mathbf{S} = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}$$

But, what if we also wanted to estimate a covariance?

Covariance structures

In our previous model, it was assumed that u_{0i} and u_{1i} are independent. That is,

$$\mathbf{S} = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}$$

But, what if we also wanted to estimate a covariance?

```
. xtmixed math5 math3 || school: math3, cov(unstructured) var mle

Mixed-effects ML regression                     Number of obs      =       887
Group variable: school                          Number of groups   =        48

                                                Obs per group: min =         5
                                                               avg =      18.5
                                                               max =        62

                                                Wald chi2(1)       =    204.24
Log likelihood = -2757.0803                     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6123977   .0428514    14.29   0.000     .5284104     .696385
       _cons |   30.34799    .374883    80.95   0.000     29.61323    31.08274
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
school: Unstructured         |
                  var(math3) |   .0343031   .0176068       .012544    .0938058
                  var(_cons) |   4.872801   1.384916      2.791615    8.505537
            cov(math3,_cons) |  -.3743092   .1273684     -.6239466   -.1246718
-----------------------------+------------------------------------------------
               var(Residual) |   26.96459   1.346082      24.45127    29.73624
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =    78.01   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference
```

Notes

- We also added options `variance` and `mle` to fully reproduce the results found in the `gllamm` manual

- Again, we can compare this model with previous using `lrtest`

- Available covariance structures are Independent (default), Identity, Exchangeable, and Unstructured

Notes

- We also added options `variance` and `mle` to fully reproduce the results found in the `gllamm` manual
- Again, we can compare this model with previous using `lrtest`
- Available covariance structures are Independent (default), Identity, Exchangeable, and Unstructured

Notes

- We also added options `variance` and `mle` to fully reproduce the results found in the `gllamm` manual
- Again, we can compare this model with previous using `lrtest`
- Available covariance structures are Independent (default), Identity, Exchangeable, and Unstructured

ML vs. REML

- ML is based on standard normal theory
- With REML, the likelihood is that of a set of linear constrasts of **y** that do not depend on the fixed effects
- REML variance components are less biased in small samples, since they incorporate degrees of freedom used to estimated fixed effects
- REML estimates are unbiased in balanced data
- LR tests are always valid with ML, not so with REML
- Very much a matter of personal taste

ML vs. REML

- ML is based on standard normal theory
- With REML, the likelihood is that of a set of linear constrasts of **y** that do not depend on the fixed effects
- REML variance components are less biased in small samples, since they incorporate degrees of freedom used to estimated fixed effects
- REML estimates are unbiased in balanced data
- LR tests are always valid with ML, not so with REML
- Very much a matter of personal taste

ML vs. REML

- ML is based on standard normal theory
- With REML, the likelihood is that of a set of linear constrasts of **y** that do not depend on the fixed effects
- REML variance components are less biased in small samples, since they incorporate degrees of freedom used to estimated fixed effects
- REML estimates are unbiased in balanced data
- LR tests are always valid with ML, not so with REML
- Very much a matter of personal taste

ML vs. REML

- ML is based on standard normal theory
- With REML, the likelihood is that of a set of linear constrasts of **y** that do not depend on the fixed effects
- REML variance components are less biased in small samples, since they incorporate degrees of freedom used to estimated fixed effects
- REML estimates are unbiased in balanced data
- LR tests are always valid with ML, not so with REML
- Very much a matter of personal taste

ML vs. REML

- ML is based on standard normal theory
- With REML, the likelihood is that of a set of linear constrasts of **y** that do not depend on the fixed effects
- REML variance components are less biased in small samples, since they incorporate degrees of freedom used to estimated fixed effects
- REML estimates are unbiased in balanced data
- LR tests are always valid with ML, not so with REML
- Very much a matter of personal taste

ML vs. REML

- ML is based on standard normal theory
- With REML, the likelihood is that of a set of linear constrasts of **y** that do not depend on the fixed effects
- REML variance components are less biased in small samples, since they incorporate degrees of freedom used to estimated fixed effects
- REML estimates are unbiased in balanced data
- LR tests are always valid with ML, not so with REML
- Very much a matter of personal taste

Two-level models

Example

Baltagi et al. (2001) estimate a Cobb-Douglas production function examining the productivity of public capital in each state's private output.

For $\mathbf{y}$ equal to the log of the gross state product measured each year from 1970-1986, the model is

$$\mathbf{y}_{ij} = \mathbf{X}_{ij}\boldsymbol{\beta} + u_i + v_{j(i)} + \epsilon_{ij}$$

for $j = 1, ..., M_i$ states nested within $i = 1, ..., 9$ regions. $\mathbf{X}$ consists of various economic factors treated as fixed effects.

Estimated fixed effects

```
. xtmixed gsp private emp hwy water other unemp || region: || state:
Mixed-effects REML regression                   Number of obs      =       816
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum
region	9	51	90.7	136
state	48	17	17.0	17

```
                                                Wald chi2(6)       =  18382.39
Log restricted-likelihood =  1404.7101          Prob > chi2        =    0.0000
```

gsp	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
private	.2660308	.0215471	12.35	0.000	.2237993	.3082624
emp	.7555059	.0264556	28.56	0.000	.7036539	.8073579
hwy	.0718857	.0233478	3.08	0.002	.0261249	.1176464
water	.0761552	.0139952	5.44	0.000	.0487251	.1035853
other	-.1005396	.0170173	-5.91	0.000	-.1338929	-.0671862
unemp	-.0058815	.0009093	-6.47	0.000	-.0076636	-.0040994
_cons	2.126995	.1574864	13.51	0.000	1.818327	2.435663

Estimated variance components

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
sd(_cons)	.0435471	.0186292	.0188287	.1007161
state: Identity				
sd(_cons)	.0802737	.0095512	.0635762	.1013567
sd(Residual)	.0368008	.0009442	.034996	.0386987

LR test vs. linear regression: chi2(2) = 1162.40 Prob > chi2 = 0.0000

Constraints on variance components

We begin by adding some random coefficients at the region level

```
. xtmixed gsp private emp hwy water other unemp || region: hwy unemp || state:,
> nolog nogroup nofetable
Mixed-effects REML regression                   Number of obs      =       816
                                                Wald chi2(6)       =  16803.51
Log restricted-likelihood =  1423.3455          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
region: Independent          |
                     sd(hwy) |   .0052752   .0108846      .0000925    .3009897
                   sd(unemp) |   .0052895    .001545       .002984    .0093764
                   sd(_cons) |   .0596008   .0758296      .0049235     .721487
-----------------------------+------------------------------------------------
state: Identity              |
                   sd(_cons) |   .0807543    .009887      .0635259    .1026551
-----------------------------+------------------------------------------------
                sd(Residual) |   .0353932    .000914      .0336464    .0372307
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(4) =  1199.67   Prob > chi2 = 0.0000
```

Constraints on variance components

We can constrain the variance components on `hwy` and `unemp` to be equal with

```
. xtmixed gsp private emp hwy water other unemp || region: hwy unemp, cov(ident
> ity) || region: || state:, nolog nogroup nofetable
Mixed-effects REML regression                   Number of obs      =       816
                                                Wald chi2(6)       =  16803.41
Log restricted-likelihood =  1423.3455          Prob > chi2        =    0.0000
```

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
region: Identity				
sd(hwy unemp)	.0052896	.0015446	.0029844	.0093752
region: Identity				
sd(_cons)	.0595029	.0318238	.0208589	.1697401
state: Identity				
sd(_cons)	.080752	.0097453	.0637425	.1023006
sd(Residual)	.0353932	.0009139	.0336465	.0372306

```
LR test vs. linear regression:      chi2(3) =  1199.67   Prob > chi2 = 0.0000
```

Factor Notation

Sometimes random effects are *crossed* rather than nested

Example

Consider a dataset consisting of weight measurements on 48 pigs at each of 9 weeks. We wish to fit the following model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij}$$

for $i = 1, ..., 48$ pigs and $j = 1, ..., 9$ weeks

Note that the week random effects v_j are not nested within pigs, they are assumed to be the same for each pig

Factor Notation

Sometimes random effects are *crossed* rather than nested

Example

Consider a dataset consisting of weight measurements on 48 pigs at each of 9 weeks. We wish to fit the following model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij}$$

for $i = 1, \ldots, 48$ pigs and $j = 1, \ldots, 9$ weeks

Note that the `week` random effects v_j are not nested within pigs, they are assumed to be the same for each pig

Factor Notation

Sometimes random effects are *crossed* rather than nested

Example

Consider a dataset consisting of weight measurements on 48 pigs at each of 9 weeks. We wish to fit the following model

$$\texttt{weight}_{ij} = \beta_0 + \beta_1 \texttt{week}_{ij} + u_i + v_j + \epsilon_{ij}$$

for $i = 1, \ldots, 48$ pigs and $j = 1, \ldots, 9$ weeks

Note that the `week` random effects v_j are not nested within pigs, they are assumed to be the same for each pig

Fitting the model

One approach to fitting this model is to consider the data as a whole and treat the random effects as random coefficients on lots of indicator variables, that is

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_{48} \\ v_1 \\ \vdots \\ v_9 \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}); \quad \mathbf{G} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_{48} & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_9 \end{bmatrix}$$

We could generate these indicator variables, but luckily `xtmixed` has factor notation to avoid this

Fitting the model

One approach to fitting this model is to consider the data as a whole and treat the random effects as random coefficients on lots of indicator variables, that is

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_{48} \\ v_1 \\ \vdots \\ v_9 \end{bmatrix} \sim N(\mathbf{0}, \mathbf{G}); \quad \mathbf{G} = \begin{bmatrix} \sigma_u^2 \mathbf{I}_{48} & \mathbf{0} \\ \mathbf{0} & \sigma_v^2 \mathbf{I}_9 \end{bmatrix}$$

We could generate these indicator variables, but luckily `xtmixed` has factor notation to avoid this

```
. xtmixed weight week || _all: R.id || _all: R.week
Mixed-effects REML regression                   Number of obs      =       432
Group variable: _all                            Number of groups   =         1
                                                Obs per group: min =       432
                                                               avg =     432.0
                                                               max =       432

                                                Wald chi2(1)       =  11516.16
Log restricted-likelihood = -1015.4214          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0578669   107.31   0.000     6.096479    6.323313
       _cons |   19.35561   .6493996    29.81   0.000     18.08281    20.62841
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
_all: Identity               |
                    sd(R.id) |   3.892648   .4141707       3.15994    4.795252
-----------------------------+------------------------------------------------
_all: Identity               |
                  sd(R.week) |   .3337581   .1611824      .1295268    .8600111
-----------------------------+------------------------------------------------
               sd(Residual)  |   2.072917   .0755915      1.929931    2.226496
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   476.10   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference
```

Some notes

- `_all` tells `xtmixed` to treat the whole data as one big panel
- `R.`*varname* is the random-effects analog of `xi`. It creates an (overparameterized) set of indicator variables, but unlike `xi`, does this behind the scenes
- When you use `R.`*varname*, covariance structure reverts to Identity
- There are alternate ways to fit this model with lower dimension
- The trick is to realize that all effects are nested within the data as a whole

Some notes

- `_all` tells `xtmixed` to treat the whole data as one big panel
- `R.`*varname* is the random-effects analog of `xi`. It creates an (overparameterized) set of indicator variables, but unlike `xi`, does this behind the scenes
- When you use `R.`*varname*, covariance structure reverts to Identity
- There are alternate ways to fit this model with lower dimension
- The trick is to realize that all effects are nested within the data as a whole

Some notes

- `_all` tells `xtmixed` to treat the whole data as one big panel
- `R.`*varname* is the random-effects analog of `xi`. It creates an (overparameterized) set of indicator variables, but unlike `xi`, does this behind the scenes
- When you use `R.`*varname*, covariance structure reverts to Identity
- There are alternate ways to fit this model with lower dimension
- The trick is to realize that all effects are nested within the data as a whole

Some notes

- `_all` tells `xtmixed` to treat the whole data as one big panel
- `R.`*varname* is the random-effects analog of `xi`. It creates an (overparameterized) set of indicator variables, but unlike `xi`, does this behind the scenes
- When you use `R.`*varname*, covariance structure reverts to Identity
- There are alternate ways to fit this model with lower dimension
- The trick is to realize that all effects are nested within the data as a whole

Some notes

- `_all` tells `xtmixed` to treat the whole data as one big panel
- `R.`*varname* is the random-effects analog of `xi`. It creates an (overparameterized) set of indicator variables, but unlike `xi`, does this behind the scenes
- When you use `R.`*varname*, covariance structure reverts to Identity
- There are alternate ways to fit this model with lower dimension
- The trick is to realize that all effects are nested within the data as a whole

```
. xtmixed weight week || _all: R.id || week:
Mixed-effects REML regression                   Number of obs      =       432

---------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+----------------------------------------------
           _all |        1        432      432.0        432
           week |        9         48       48.0         48
---------------------------------------------------------------

                                                Wald chi2(1)       =  11516.16
Log restricted-likelihood = -1015.4214          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0578669   107.31   0.000     6.096479    6.323313
       _cons |   19.35561   .6493996    29.81   0.000     18.08281    20.62841
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
_all: Identity               |
                    sd(R.id) |   3.892648   .4141707       3.15994    4.795252
-----------------------------+------------------------------------------------
week: Identity               |
                   sd(_cons) |   .3337581   .1611824      .1295268    .8600112
-----------------------------+------------------------------------------------
                sd(Residual) |   2.072917   .0755915      1.929931    2.226496
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   476.10   Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference
```

```
. xtmixed weight week || _all: R.week || id:

Mixed-effects REML regression                   Number of obs      =       432

--------------------------------------------------------------------------
                |   No. of       Observations per Group
 Group Variable |   Groups    Minimum    Average    Maximum
----------------+---------------------------------------------------------
           _all |        1        432      432.0        432
             id |       48          9        9.0          9
--------------------------------------------------------------------------

                                                Wald chi2(1)       =  11516.16
Log restricted-likelihood = -1015.4214          Prob > chi2        =    0.0000

------------------------------------------------------------------------------
      weight |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        week |   6.209896   .0578669   107.31   0.000     6.096479    6.323313
       _cons |   19.35561   .6493996    29.81   0.000     18.08281    20.62841
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.     [95% Conf. Interval]
-----------------------------+------------------------------------------------
_all: Identity               |
                 sd(R.week)  |   .3337581   .1611824      .1295268    .8600112
-----------------------------+------------------------------------------------
id: Identity                 |
                  sd(_cons)  |   3.892648   .4141707       3.15994    4.795252
-----------------------------+------------------------------------------------
               sd(Residual)  |   2.072917   .0755915      1.929931    2.226496
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(2) =   476.10   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference
```

A glimpse at the future

You can welcome Stata to the game. We hope you like the syntax and output.

Some things to look for in future versions

- Correlated errors and heteroskedasticity
- Exploiting matrix sparsity/very large problems
- Factor variables
- Degrees of freedom calculations
- Generalized linear mixed models. Adding `family()` and `link()` options to what we have here

A glimpse at the future

You can welcome Stata to the game. We hope you like the syntax and output.

Some things to look for in future versions

- Correlated errors and heteroskedasticity
- Exploiting matrix sparsity/very large problems
- Factor variables
- Degrees of freedom calculations
- Generalized linear mixed models. Adding `family()` and `link()` options to what we have here

A glimpse at the future

You can welcome Stata to the game. We hope you like the syntax and output.

Some things to look for in future versions

- Correlated errors and heteroskedasticity
- Exploiting matrix sparsity/very large problems
- Factor variables
- Degrees of freedom calculations
- Generalized linear mixed models. Adding `family()` and `link()` options to what we have here

A glimpse at the future

You can welcome Stata to the game. We hope you like the syntax and output.

Some things to look for in future versions

- Correlated errors and heteroskedasticity
- Exploiting matrix sparsity/very large problems
- Factor variables
- Degrees of freedom calculations
- Generalized linear mixed models. Adding `family()` and `link()` options to what we have here

A glimpse at the future

You can welcome Stata to the game. We hope you like the syntax and output.

Some things to look for in future versions

- Correlated errors and heteroskedasticity
- Exploiting matrix sparsity/very large problems
- Factor variables
- Degrees of freedom calculations
- Generalized linear mixed models. Adding `family()` and `link()` options to what we have here

A glimpse at the future

You can welcome Stata to the game. We hope you like the syntax and output.

Some things to look for in future versions

- Correlated errors and heteroskedasticity
- Exploiting matrix sparsity/very large problems
- Factor variables
- Degrees of freedom calculations
- Generalized linear mixed models. Adding `family()` and `link()` options to what we have here