Multilevel Models

1. Introduction. Variance Components

Germán Rodríguez

Princeton University

March 26, 2018

Website

The course has a website at **https://data.princeton.edu/pop510**, where you will find supporting materials including

- a course syllabus and bibliography, with useful links to other resources
- a collection of computing logs including
  - Stata and R logs fitting various linear and generalized linear multilevel models by maximum likelihood
  - Computing logs illustrating the use of Bayesian methods in multilevel analysis, including a random-effects logistic regression model fitted using WinBUGS and Stan
  - Some older runs using the classic package MLwiN

Some of my own research on multilevel models is housed at **https://data.princeton.edu/multilevel**. Resources include a list of publications, the simulated data used in a 1995 JRSS-A paper, and the actual data used in a 2001 JRSS-A paper (and earlier in a 1996 Demography paper)

Outline

1. Introduction. Variance-component models. Maximum likelihood and empirical Bayes estimates. Random intercepts.
2. Random slopes. Contextual predictors and cross-level interactions. Longitudinal data. Growth curve models. The general linear mixed model.
3. Models for binary data and the generalized linear mixed model. Likelihood approximations (MQL and PQL). Quadrature and adaptive quadrature.
4. Bayesian estimation in multilevel models. Markov chain Monte Carlo (MCMC). Gibbs sampling. The Metropolis algorithm. Hamiltonian Monte Carlo.
5. Models for categorical data. Ordered logits. Multinomial logits via SEM and Stan. Poisson models for count data. Small area estimation.
6. Multilevel survival models. Frailty models. Relationship with generalized linear mixed models. Interpreting results.

Software

The course will emphasize applications. Software used will be

- Stata, with `xtreg`, `xtlogit` and `xtpoisson` for random-intercept models and `mixed`, `melogit` and `mepoisson` for more general multilevel models,
- R's `lme4` package with the functions `lmer()` and `glmer()`,
- WinBUGS for Bayesian inference using the Gibbs sampler and Stan for Hamiltonian MCMC using NUTS.

Other specialized software in common use includes

- MLwinN, developed by Goldstein and collaborators at the Centre for Multilevel Modelling at Bristol University, and
- HLM, developed by Raudenbush and collaborators at the University of Michigan and distributed by Scientific Software International (SSI). A free student edition is available

Bibliography

The closest thing to a course textbook is

- Rabe-Hesketh, S., and A. Skrondal. (2012). *Multilevel and Longitudinal Modeling using Stata*, 3rd edition. Volume I: Continuous Responses and Volume II: Categorical Responses, Counts, and Survival. Stata Press.

The online syllabus cites other sources. Of particular note are

- Goldstein, H. (2003). *Multilevel Statistical Models*, 3rd edition. London: Edward Arnold. 4th edition is print on demand. 2nd edition is available free in electronic form
- Gelman, A., and J. Hill. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, an excellent book on statistical modeling including multilevel models

Bates (2010) has posted chapters from a new book on mixed models with R.

Variance components

We start by considering a simple situation: 2-level clustered or longitudinal data with no covariates. The online example concerns children nested in schools.

Let Y_{ij} denote the outcome for the jth member of the ith group. (Notation is not consistent, MALMUS uses the subscripts in the reverse order. Levels may be counted up or down, and some count only grouping levels!)

The model is

$$Y_{ij} = \mu + a_i + e_{ij}$$

where μ is an overall mean, $a_i \sim N(0, \sigma_a^2)$ is a random effect for group i and $e_{ij} \sim N(0, \sigma_e^2)$ is the usual individual error term, independent of a_i.

Expectation and variance

The expected outcome in this model is just

$$E(Y_{ij}) = \mu$$

The variance of the outcome has two components

$$\text{var}(Y_{ij}) = \sigma_a^2 + \sigma_e^2$$

The covariance between two outcomes in the same group is

$$\text{cov}(Y_{ij}, Y_{ik}) = \sigma_a^2, \quad j \neq k$$

The correlation between two observations in the same group is

$$\rho = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2},$$

and is called the *intraclass* correlation.

Estimation of the mean

The OLS estimator of μ is the grand mean, which can be written as a weighted average of the group means

$$\bar{Y} = \sum_i n_i \bar{Y}_i / n$$

with weights proportional to sample sizes. This estimator is consistent but not fully efficient with unbalanced data.

The ML estimator of μ given the variances σ_a^2 and σ_e^2 is also a weighted average of the group means

$$\hat{\mu} = \sum_i w_i \bar{Y}_i / \sum w_i$$

but with weights w_i inversely proportional to their variances $\text{var}(\bar{Y}_i) = \sigma_a^2 + \sigma_e^2/n_i$, for which we plugin estimates.

Estimation of the variances

There are three approaches to estimating the variance components

- A method of moments or anova estimator equates the between-groups and within-groups sums of squares to their expected values.
- The maximum likelihood (ML) estimator maximizes the multivariate normal likelihood of the data with respect to all parameters, implemented in Stata and R.
- The restricted maximum likelihood (REML) estimator uses the likelihood of error contrasts, which allow for the estimation of the fixed parameters (analogous to using $n - p$ instead of n in OLS), also implemented in Stata and R.

Each of these approaches leads to a (usually slightly) different estimator of the mean. I generally prefer ML because it allows likelihood ratio tests for nested models.

Scores in language tests

For the language data in the computing logs the MLE of the mean is

$$\hat{\mu} = 40.364$$

compared to a grand mean of 40.935.
The estimated variance components are

$$\hat{\sigma}_a^2 = 4.408^2 \quad \text{and} \quad \hat{\sigma}_e^2 = 8.035^2$$

The intraclass correlation, or correlation between the scores of two students in the same school, is

$$\hat{\rho} = \frac{4.408^2}{4.408^2 + 8.035^2} = 0.231$$

so 23% of the variation in language scores can be attributed to the schools.

Tests of hypotheses

To test hypotheses about μ we used the fact that the ML (or REML) estimate is asymptotically normal, so we can use a Wald test. In particular, we can compute a 95% confidence interval

$$\hat{\mu} \pm 1.96\, \hat{\text{se}}(\hat{\mu})$$

For the school data the 95% interval for the mean is $(39.52, 41.20)$

To test hypotheses about σ_a^2 we can use a likelihood ratio test, fitting models with and without school effects. Because the hypothesis is on a boundary of the parameter space, however, the test criterion does not have the usual χ_1^2 distribution, but is best treated as a 50:50 mixture of 0 and χ_1^2 by halving the p-value.

For the school data the test criterion is $\bar{\chi}_{01}^2 = 287.98$, so we have highly significant school effects on language scores.

Predicting the random effects - ML

Consider "estimating" the school means

$$E(Y_{ij}|a_i) = \mu + a_i$$

Because these involve the random variables a_i we prefer to use the term "prediction". MALMUS uses "assigning numbers".

One approach is to treat the a_i as if they were fixed parameters and everything else as known, and then use ML. The resulting estimate is the difference between the group mean and the MLE of μ

$$\hat{a}_i^{ML} = \bar{y}_i - \hat{\mu}$$

Thus the ML estimate of the school mean is the sample mean $\bar{y}_i$.

Predicting the random effects - BLUP

An alternative approach is to minimize the prediction variance using a best linear unbiased predictor (BLUP), which has the form

$$\hat{a}_i^{BLUP} = \hat{a}_i^{ML} \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2/n_i}$$

The fraction on the right "shrinks" the ML towards zero by an amount that depends on the reliability of $\bar{y}_i$.

The corresponding estimator of the school mean is a compromise between the sample school mean and the overall mean

$$\hat{\mu}_i = \bar{y}_i w_i + \hat{\mu}(1 - w_i)$$

where w_i is the fraction in the top equation. These estimators have an empirical Bayes interpretation which we discuss next.

Bayes theorem

Bayes theorem gives us the conditional probability of $P|D$ as a function of the conditional probability of $D|P$

$$\Pr(P|D) = \frac{\Pr(PD)}{\Pr(D)} = \frac{\Pr(D|P)\Pr(P)}{\Pr(D)}$$

Suppose P are the parameters (which Bayesians view as random) and D are the data, so $\Pr(D|P)$ is the usual likelihood, $\Pr(P)$ is called the prior, and $\Pr(P|D)$ is the posterior distribution, so we can write

$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

Here we ignored $\Pr(D)$, which is just a normalizing constant.

Classical statisticians base their inferences on the likelihood, while Bayesians use the posterior. If the prior is vague or uninformative the two approaches give similar results.

Empirical Bayes

In the variance components model we treat a_i as a parameter with prior $N(0, \sigma_a^2)$.

The likelihood is the distribution of the $Y_{ij}|a_i$ for $j = 1, \ldots, n_i$ which are independent $N(\mu + a_i, \sigma_e^2)$. Maximizing yields $\hat{a}_i^{ML}$.

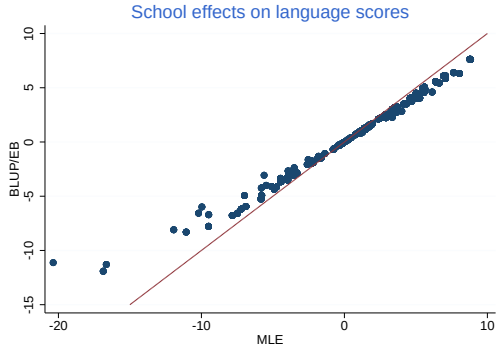
The posterior or distribution of $a_i|y_{ij}$ is proportional to the product of the prior and likelihood and can be shown to be a normal distribution, so the mean and mode coincide and give $\hat{a}_i^{EB}$.

This is not a full Bayesian approach, because instead of assuming priors for μ, σ_a^2 and σ_e^2 we simply plugged estimates. Hence the name *empirical* Bayes (EB).

This approach can be used to produce demographic estimates for small areas, where sample means may be unreliable

Shrinking towards zero

In the computing logs we show how to fit the variance-components model and predict the school random effects using ML and EB. The following graph shows substantial "shrinkage" for three small schools.



Multilevel Models

2. Random Intercept Models

Germán Rodríguez

Princeton University

March 28, 2018

Random intercepts

We now consider models with covariates, starting with the random-intercept model

$$Y_{ij} = \alpha + a_i + x'_{ij}\beta + e_{ij}$$

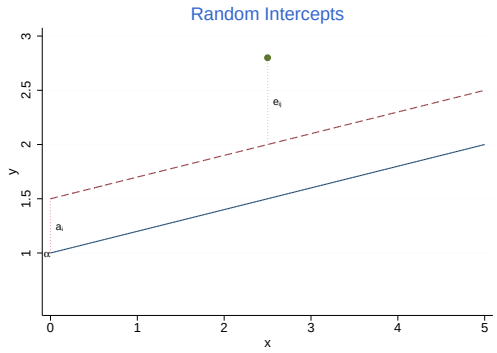
where Y_{ij} is the outcome for the j-th individual in the i-th group, α is the constant, and x_{ij} is a vector of predictors with coefficients β.

We have two residuals: a group random effect $a_i \sim N(0, \sigma_a^2)$ and an individual effect $e_{ij} \sim N(0, \sigma_e^2)$, assumed independent of each other and of the covariates.

Given the random effect a_i, the outcome $Y_{ij}|a_i$ follows an ordinary linear model with intercept $\alpha + a_i$, hence the name "random intercept".

Parallel lines

Here's the model in graphical form



In terms of our example, we assume that language scores depend on verbal IQ with a common slope and an intercept that varies across schools.

Unconditional moments

In this model the expected value of the outcome is

$$E(Y_{ij}) = \alpha + x'_{ij}\beta$$

The variances and covariances of the outcomes are

$$\text{var}(Y_{ij}) = \sigma_a^2 + \sigma_e^2 \quad \text{and} \quad \text{cov}(Y_{ij}, Y_{ik}) = \sigma_a^2, j \neq k$$

The correlation between any two outcomes in a group, or *intraclass correlation* is

$$\rho = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2}$$

These results are exactly the same as in the variance-components model, the only difference is that we now account for covariates.

Estimation of the parameters

The OLS estimator of α and β ignoring the correlation structure is consistent but not fully efficient. The associated standard errors need to be corrected for clustering.

A better approach is to use maximum likelihood (ML). This is implemented in Stata's `xtreg, mle` and `mixed, mle`, as well as R's `lmer()` if you specify `REML = FALSE`.

Alternatively, one can use restricted maximum likelihood (REML), as implemented in Stata's `mixed, reml`, or as the default in R's `lmer()`, which relies on error contrasts to estimate the variance components.

Given estimates of σ_a^2 and σ_e^2, both ML and REML estimate α and β using generalized least squares. The two procedures give very similar estimates if the number of groups is large.

Language scores

The computing logs fit this model to the language score data as a function of verbal IQ centered on the grand mean, so the model is

$$Y_{ij} = \alpha + a_i + \beta(x - \bar{x}_i) + e_{ij}.$$

The fitted equation is

$$E(Y_{ij}) = 40.609 + 2.488(x - \bar{x})$$

We estimate the variances as

$$\hat{\sigma}_a^2 = 3.082^2 \quad \text{and} \quad \hat{\sigma}_e^2 = 6.498^2$$

The intraclass correlation is

$$\hat{\rho} = \frac{3.082^2}{3.082^2 + 6.498^2} = 0.167,$$

so schools account for 17% of the variation in language scores after taking into account verbal IQ.

Multilevel R^2

In ordinary linear models we compute R^2 for model ω as the proportionate reduction in the RSS starting from the null model ϕ

$$R^2 = 1 - \frac{\text{RSS}(\omega)}{\text{RSS}(\phi)}$$

In a two-level random intercept model we can define R^2 as

$$R^2 = 1 - \frac{\hat{\sigma}_a^2(\omega) + \hat{\sigma}_e^2(\omega)}{\hat{\sigma}_a^2(\phi) + \hat{\sigma}_e^2(\phi)} = 0.384$$

This statistic can also be calculated by level

$$R_a^2 = 1 - \frac{\hat{\sigma}_a^2(\omega)}{\hat{\sigma}_a^2(\phi)} = 0.511 \quad \text{and} \quad R_e^2 = 1 - \frac{\hat{\sigma}_e^2(\omega)}{\hat{\sigma}_e^2(\phi)} = 0.346$$

Unlike linear models R^2 is not guaranteed to increase when variables are added!

Predicting random effects

Consider now estimating the group intercepts $\alpha + a_i$, which involves predicting a_i given the other parameters in the model

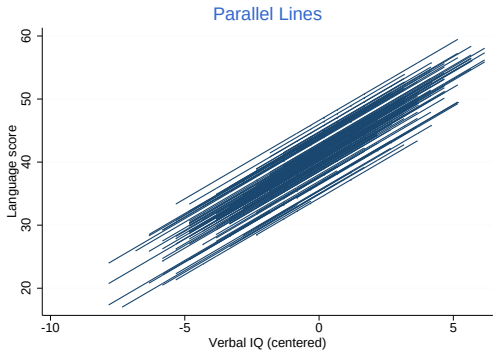
The ML estimator of a_i is obtained by treating everything else as known and maximizing the likelihood, and turns out to be the group means of the residuals $y_{ij} - (\hat{\alpha} + x_{ij}'\hat{\beta})$.

The EB estimator maximizes the posterior distribution, obtained as the product of the likelihood and prior, and can be obtained in Stata using `predict, reffects` after `mixed` (but not after `xtreg`), and in R via `ranef()`.

Comparing the EB and ML estimators you should expect the usual shinkrage towards zero, as we'll soon see.

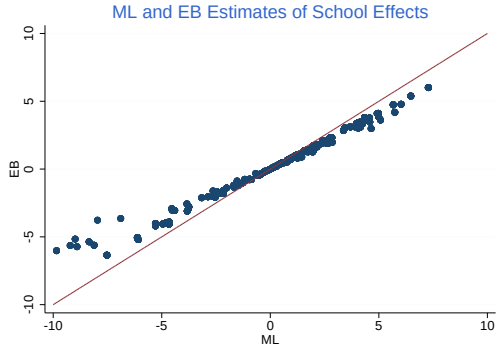
Parallel lines

In the computing logs we fit the model using maximum likelihood and then obtain the fitted values $\hat{y}_{ij} = \hat{\alpha} + \hat{a}_i + x'_{ij}\hat{\beta}$. The figure below shows these lines.



Comparison of ML and EB

We can also compare the empirical Bayes estimates with the maximum likelihood estimates of the school effects.



The computing logs have a similar plot using estimated intercepts rather than school effects.

Hypothesis testing

To test hypotheses about β, for example the hypothesis $H_0 : \beta_2 = 0$ for a subset β_2, we can use

1. Wald tests, constructing the quadratic form $W = \hat{\beta}_2' \text{var}(\hat{\beta}_2)^{-1}\hat{\beta}_2$, which is asymptotically χ^2_p with d.f. equal to the number of coefficients in β_2.
2. Likelihood ratio tests, where we fit the model with and without the predictors involved in β_2 and take twice the difference in log-likelihoods, which is asymptotically χ^2_p with the same d.f. as above.

With ML both tests are available. With REML we can only use Wald tests; because the models with and without β_2 use different error contrasts the restricted likelihoods are not nested!

Tests about the variance components proceed as before.

Language scores

For the language scores data the estimated slope of 2.49 has a standard error of 0.07 leading to a Wald z=35.5 (equivalent to $\chi^2_1 = 1261.4$) and a 95% confidence interval of $(2.35, 2.63)$.

The likelihood ratio test compares the log-likelihoods with and without verbal IQ and gives $\chi^2_1 = 1001.5$. In multilevel linear models the LR and Wald tests are $\neq$, but asympt equivalent.

The LR test is not available if you use REML, which used to be the default of `mixed`. If you try, Stata will warn "REML criterion is not comparable under different fixed-effects specifications"

Tests for the variance components are as before. In our example the test for σ^2_a is $\bar{\chi}_{01} = 225.92$ using ML and $\bar{\chi}_{01} = 227.30$ using REML. School effects are clearly significant.

Between-groups estimation

Consider the group means, $\bar{Y}_i = \sum_j Y_{ij}/n_i$, which follow the model

$$\bar{Y}_i = \alpha + a_i + \bar{X}_i'\beta + \bar{e}_i$$

where $\bar{X}_i$ is the average of the covariates and $\bar{e}_i \sim N(0, \sigma_e^2/n_i)$ is the average error term.

These means are independent and we can estimate α and β by OLS or WLS.

Stata can compute this estimator via the command `xtreg, be`. The option `wls` uses group sizes as weights. (Ideally, of course, we would like to use weights inversely proportional to the variances of the group means.)

The between-groups estimator of the slope for the language score data using WLS is 3.90, much larger than the RE estimate of 2.49.

Within-groups estimation

Consider now the differences between the individual outcomes and the group means. These follow the model

$$Y_{ij} - \bar{Y}_i = (X_{ij} - \bar{X}_i)'\beta + (e_{ij} - \bar{e}_i)$$

Note that α drops out, as would any variables which are constant within groups. (None in our example.)

The estimator based on within-group variation is known as the **fixed effects** estimator, and is equivalent to using a dummy variable for each group. It is available in Stata in `xtreg, fe`, and in R using the package `plm`.

The within-groups estimate of the IQ slope for the language score data is 2.41.

Between and within-groups together

The random effects estimator is a weighted average of the between and within estimators.
It is possible to obtain the within and between-groups estimates together by fitting a model that includes as predictors the group means and the differences from the group means:

$$y_{ij} = \alpha + a_i + \bar{x}_i\beta_B + (x_{ij} - \bar{x}_i)\beta_W + e_{ij}$$

For the language score data we obtain estimates of

$$\hat{\beta}_B = 4.00 \quad \text{and} \quad \hat{\beta}_W = 2.41$$

The between estimator differs slightly from the WLS estimator because the weights are not the exactly the same. The within estimator is identical to the fixed-effects estimator.

A Wald test of equality gives $\chi^2_1 = 25.79$ and casts doubt on the validity of the random effects estimator.

The Hausman specification test

Hausman has proposed a specification test for the random effects model and the assumption that the school effects are exogenous

If the group effects are in fact independent of observed covariates then the random effects estimator is both consistent and efficient.

If the group effects are correlated with observed covariates, then the fixed effects estimator is consistent but not efficient.

The Hausman test is based on the quadratic form

$$(\hat{\beta}_E - \hat{\beta}_C)'[\widehat{\text{var}}(\hat{\beta}_C) - \widehat{\text{var}}(\hat{\beta}_E)]^{-1}(\hat{\beta}_E - \hat{\beta}_C)$$

where I used E for efficient (here the random-effects estimator) and C for consistent (here the fixed-effects).

For the language score data the Hausman test gives $\chi^2_1 = 33.75$, strong evidence of model misspecification.

Fixed or random?

A random-effects model that fails the Hausman test is often abandoned in favor of the fixed-effects model. A few caveats:

- If the test rules out RE it doesn't mean FE is the correct model! Both models assume uncorrelated errors at level 1. It is only omitted variables at level 2 that are handled by FE.
- Using FE precludes estimating coefficients for variables that are constant within a group (for example school SES). Sometimes these effects are of primary interest.
- An alternative approach is to include the group means as predictors for any variable where the between and within group estimators are significantly different.

Centering on group means

Centering the individual predictors on the group means is *optional* when the group means are included in the model.
To see this point write

$$\bar{x}_i'\beta_B + (x_{ij} - \bar{x}_i)'\beta_W = \bar{x}_i'(\beta_B - \beta_W) + x_{ij}'\beta_W$$

So all that happens if x_{ij} is not centered is that the coefficient of $\bar{x}_i$ becomes the *difference* between the between and within coefficients, which is convenient for testing.

Centering the individual predictors on the group means does not make a lot of sense if the group means are not included in the model. Centering on the grand mean is fine in all models that include a constant.

Ignoring clustering

What happens if the random-intercept model is correct but we ignore the clustering?

- As noted earlier, the OLS estimate of the fixed effects is consistent but not fully efficient, so this is not a serious problem in large samples
- The estimated standard errors, however, are incorrect. A common misconception is that they are always too small. As shown in MALMUS §3.10
  - They are too small for between-cluster covariates
  - but too large for within-cluster covariates

  The solution is to adjust for clustering or, better still, use ML.

Exercise. Verify the last statement using the language scores in the website, fitting a model with verbal IQ and school SES as examples of within and between predictors using OLS and ML.

Multilevel Models

3. Random Coefficients

Germán Rodríguez

Princeton University

April 2, 2018

Group-specific regressions

We return to our analysis of language scores by verbal IQ in 131 schools in the Netherlands.

A simple approach to the analysis of the data would fit separate regressions in each school using the model

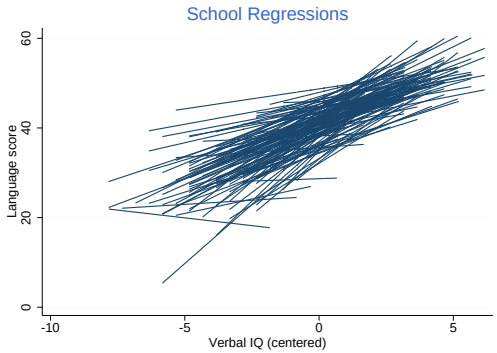
$$Y_{ij} = \alpha_i + \beta_i x_{ij} + e_{ij}$$

where $e_{ij} \sim N(0, \sigma_e^2)$, fitted just to school i (so in fact the error variance could vary across schools).

This is easy to do as shown in the computing logs, where we use the `statsby` command in Stata and `dplyr`'s `group_by()` in R, to run a simple linear regression in each school and gather the intercepts and slopes.

School regression lines

We can then plot the 131 regression lines



Some of these lines are based on relatively small schools and are thus rather poorly estimated.

Random-coefficient models

An alternative approach is to view the intercept and slope of the regression lines as random:

$$Y_{ij} = (\alpha + a_i) + (\beta + b_i)x_{ij} + e_{ij}$$

where α is the average intercept and β the average slope, a_i is a school effect on the intercept and b_i is a school effect on the slope, and e_{ij} is the usual error term, with $e_{ij} \sim N(0, \sigma_e^2)$.

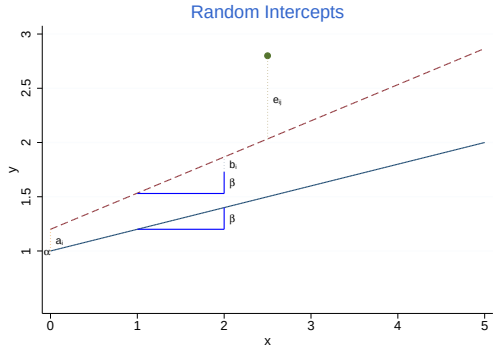
The distribution of the school effects on the intercept and slope is *bivariate* normal with mean zero and a general variance-covariance

$$\begin{pmatrix} a_i \\ b_i \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_a^2 & \sigma_{ab} \\ \sigma_{ab} & \sigma_b^2 \end{pmatrix}\right),$$

that allows for correlation between the intercept and slope effects.

Random Slopes

The figure below shows the model in graphical form



Interpretation of the intercept (and its variance) depends on the predictor and is more natural if the predictor is centered.

Moments and variances

In this model the expected outcome is a linear function of x_{ij} reflecting the average or pooled regression line

$$E(Y_{ij}) = \alpha + \beta x_{ij}$$

The variance of the outcome turns out to be

$$\text{var}(Y_{ij}) = \sigma_a^2 + 2x_{ij}\sigma_{ab} + x_{ij}^2\sigma_b^2 + \sigma_e^2$$

and depends on x_{ij}, so the model is *heteroscedastic*.

The covariance between two outcomes in the same group, is

$$\text{cov}(Y_{ij}, Y_{ik}) = \sigma_a^2 + (x_{ij} + x_{ik})\sigma_{ab} + x_{ij}x_{ik}\sigma_b^2,\ j \neq k$$

and depends on the two values of the predictor, x_{ij} and x_{ik}.

Intraclass correlation

As a result, the intraclass correlation is now a function of the covariates! With one predictor

$$\rho_{i\,jk} = \frac{\sigma_a^2 + (x_{ij} + x_{ik})\sigma_{ab} + x_{ij}x_{ik}\sigma_b^2}{\sqrt{\sigma_a^2 + 2x_{ij}\sigma_{ab} + x_{ij}^2\sigma_b^2 + \sigma_e^2}\sqrt{\sigma_a^2 + 2x_{ik}\sigma_{ab} + x_{ik}^2\sigma_b^2 + \sigma_e^2}}$$

To obtain a single-number summary we can compute the intraclass correlation for 'average' individuals

$$\bar{\rho} = \frac{\sigma_a^2 + 2\bar{x}\sigma_{ab} + \bar{x}^2\sigma_b^2}{\sigma_a^2 + 2\bar{x}\sigma_{ab} + \bar{x}^2\sigma_b^2 + \sigma_e^2}$$

where $\bar{x}$ is the overall mean of the predictor.

If the predictor is centered we obtain a more familiar result

$$\bar{\rho} = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2}$$

Language scores: model

The computing logs show how to fit a random-slope model to the data on language scores by verbal IQ in Stata and R. In Stata the command is

```
mixed langpost iqvc || schoolnr:  iqvc, ///
      mle covariance(unstructured)
```

The two vertical bars separate the fixed and random parts of the model. We treat the constant and verbal IQ as random at the school level. It is important to request an *unstructured* covariance matrix, otherwise Stata assumes independence.

In R we call the `lmer()` function with a two-sided formula

```
lmer(langpost ~ iqvc + (iqvc | schoolnr),
     data = snijders, REML = FALSE)
```

Here it is important to request ML as the default in R is REML.

Language scores: estimates

The average relationship across schools is

$$E(Y_{ij}) = 40.71 + 2.53(x - \bar{x})$$

but we find substantial variation across schools in both the intercept and slope, with

$$\hat{\sigma}_a^2 = 3.056^2 \quad \text{and} \quad \hat{\sigma}_b^2 = 0.458^2$$

There's also a negative correlation between the intercept and slope (not to be confused with the intraclass correlation)

$$r_{ab} = -0.817$$

The large negative correlation is not unusual, which is why it is important to use an unstructured covariance matrix.

Finally, the error variance is estimated as

$$\hat{\sigma}_e^2 = 6.44^2$$

Language scores: intraclass correlation

The correlation between the language scores of two students with average verbal IQ in the same school is

$$\hat{\rho}(\bar{x}) = \frac{3.06^2}{3.06^2 + 6.44^2} = 0.184$$

which also means that 18.4% of the variance in language scores at average IQ occurs between schools.

Exercise: calculate the intraclass correlation for different values of verbal IQ and plot it.

You should find that the correlation in language scores for children in the same school who have verbal IQs one standard deviation below the overall mean is 0.279. What happens above the mean?

Testing Hypotheses

Tests of hypotheses proceed along the same general lines as in random-intercept models. Here are the highlights:

- We can test the significance of the average effect of verbal IQ using a Wald test. We get $z = 31.0$, equivalent to $\chi^2_1 = 962.0$.
- We can test the significance of the variance of the slope across schools using a likelihood ratio test. Removing verbal IQ from the random part saves two parameters. Equivalently, $\sigma_a^2 = 0$ implies $\sigma_{ab} = 0$, so we test that both are zero. We get $\chi^2 = 21$ using ML. The test can also be done using REML.
- Because the above test is on a boundary of the parameter space it does not have the usual χ^2_2 distribution. Stata treats it as conservative, whereas MALMUS recommend using a 50:50 mixture of χ^2_1 and χ^2_2. Either way it is highly significant.
- Removing verbal IQ from the fixed and random parts of the model saves 3 parameters. The LR test is conservatively χ^2_3.

Estimating intercepts and slopes

The ML estimates of a_i and b_i can be obtained by calculating the residuals from the fixed part of the model

$$y_{ij} - (\hat{\alpha} + \hat{\beta} x_{ij})$$

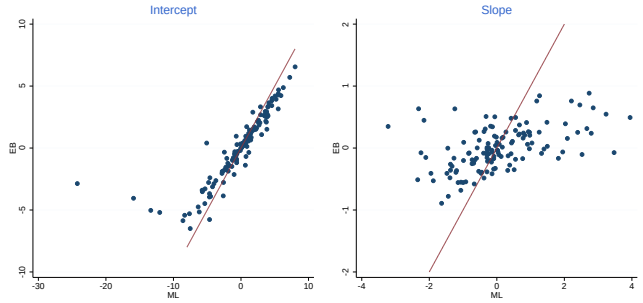
and then fitting school-specific regressions of those residuals on x_{ij} by OLS.

The EB estimates can be obtained in Stata by using `predict bi ai, reffects` after `mixed` (warning: Stata outputs slopes *before* intercepts) and in R by calling `ranef()`.

Both methods treat the fixed effects as known and substitute estimates. Typically the EB estimates show shrinkage towards zero compared to the ML estimates.

ML and EB estimates

The following figure compares ML and EB estimates of the school effects on the intercept and slope for the language score data

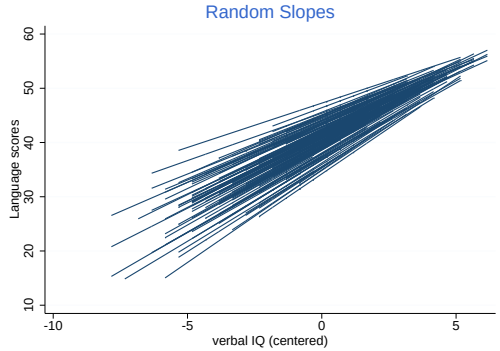


We see substantial shrinkage of the intercept effects for a few small schools, and of the slope effects all around.

Note: The online log has very similar plots comparing empirical Bayes with school-specific regressions

Predicted regression lines

We can also plot the predicted regression lines for all schools



The figure shows that school differences in language scores are more pronounced at low verbal IQs than at the high end.

Level-2 predictors

We now consider introducing a level-2 predictor z_i, illustrated by school SES. As usual I center the predictor on the overall mean, but will leave that implicit to simplify the notation.

We can introduce school SES as a main effect

$$Y_{ij} = (\alpha + a_i) + (\beta + b_i)x_{ij} + \gamma z_i + e_{ij}$$

We can also interact school SES with student verbal IQ

$$Y_{ij} = (\alpha + a_i) + (\beta + b_i)x_{ij} + \gamma z_i + \delta x_{ij} z_i + e_{ij}$$

The additional term is called a *cross-level* interaction.

Estimation and testing proceeds as usual, with main effects and interactions in the fixed part interpreted in the standard way.

Level-specific models

There's another way to think about random-coefficient models. We start with a level-1 model

$$Y_{ij} = A_i + B_i x_{ij} + e_{ij}$$

where we predict language scores as a function of verbal IQ with coefficients that vary by school plus an error term.

We then add a level-2 model for the coefficients

$$A_i = \alpha + a_i, \quad \text{and}$$
$$B_i = \beta + b_i$$

In this case the intercept and slope are viewed as just a constant plus a residual; in other words we have *null* models at level 2.

Substituting the second set of equations on the first yields the random-slope model in slide 4.

Level-2 models

We now introduce school level SES as a predictor in the level-2 equations, writing

$$A_i = \alpha_0 + \alpha_1 z_i + a_i, \quad \text{and}$$
$$B_i = \beta_0 + \beta_1 z_i + b_i$$

where the level-2 residuals a_i and b_i have a bivariate normal distribution with mean zero and unstructured covariance matrix.

In this model both the intercept and slope are linear functions of school SES, but we could let just the intercept depend on SES.

Substituting these equations in the original model gives

$$Y_{ij} = (\alpha_0 + \alpha_1 z_i + a_i) + (\beta_0 + \beta_1 z_i + b_i)x_{ij} + e_{ij}$$

and rearranging terms leads back to the model with a cross-level interaction in slide 15.

Level-specific and reduced forms

The package HLM uses the level-specific formulation of the model, whereas Stata and R use the reduced form and requires the user to specify the cross-level interaction terms, but the models are equivalent and the estimates are exactly the same.

MALMUS notes that users of HLM tend to include more cross-level interactions than users of Stata because they are built-in.

In the computing logs we fit a model using centered verbal IQ, centered school SES, and the interaction term. In Stata

```
mixed langpost iqvc sesc iqvcXsesc ///
|| schoolnr:  iqvc, mle covariance(unstructured)
```

The specification of the random part is exactly the same as before we introduced SES, but the fixed part now has the main effect of SES and its cross-level interaction with verbal IQ.

Fixed effects

Here are the ML estimates of the fixed effects

```
                                        Wald chi2(3)      =   1059.58
Log likelihood = -7607.8383             Prob > chi2       =    0.0000

------------------------------------------------------------------------------
    langpost |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        iqvc |   2.515011   .0789096    31.87   0.000     2.360351    2.669671
        sesc |   .2410466   .0658997     3.66   0.000     .1118856    .3702077
   iqvcXsesc |  -.0470625   .0174818    -2.69   0.007    -.0813263   -.0127988
       _cons |   40.71326   .2910762   139.87   0.000     40.14276    41.28376
------------------------------------------------------------------------------
```

These can be written as the following estimated equation

$$E(Y_{ij}|a_i, b_i) = (40.713 + 0.241 z_i + a_i) + (2.515 - 0.047 z_i + b_i)x_{ij}$$

The expected score for an average student in the average school is 40.71, it is 1.07 higher if school SES is one sd more, 5.20 higher when verbal IQ is one sd more, and 5.84 higher if both conditions obtain. Clearly, verbal IQ makes more of a difference in schools with low SES.

Note: Verbal IQ has sd=2.07 and school SES has sd=4.43

Random effects

Here are the parameters for the random part

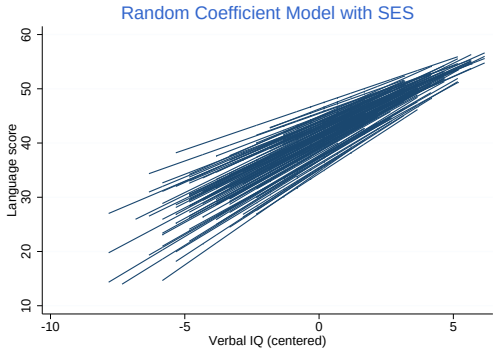
```
------------------------------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.      [95% Conf. Interval]
-----------------------------+------------------------------------------------
schoolnr: Unstructured       |
                   sd(iqvc) |   .3859035   .1208045      .2089373    .7127571
                  sd(_cons) |    2.86771   .2384242      2.436495    3.375242
             corr(iqvc,_cons) |  -.8008473   .2294731     -.9821521    .1519006
-----------------------------+------------------------------------------------
                sd(Residual) |   6.443356   .1005435      6.249277    6.643462
------------------------------------------------------------------------------
LR test vs. linear regression:       chi2(3) =   215.75   Prob > chi2 = 0.0000

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

The estimates are similar to the previous model, with substantial effects of unobserved school characteristics after accounting for school SES. These effects exceed those of SES.

The intraclass correlation for average students in the average school is 0.165, and now represents variation in the scores of average students across schools with the same SES.

Predicted school lines

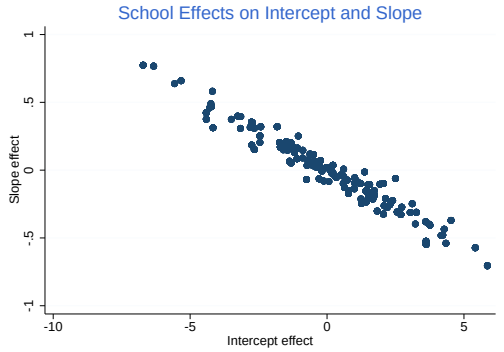
We can calculate EB estimates of the random effects as usual and plot the predicted school-level regressions



The figure looks very similar to the previous analysis, with smaller school differences at higher verbal IQs, all at observed SES.

Empirical Bayes estimates

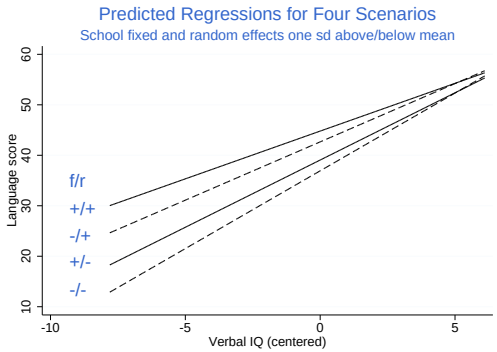
We can also plot the empirical Bayes estimates of the school effects on the intercept and slope



The prior correlation is -0.801 and the posterior correlation is -0.971. Schools with higher language scores at average verbal IQ show smaller differences by verbal IQ.

Observed and unobserved effects

To compare observed and unobserved school effects we look at four school scenarios, setting SES one sd above/below the mean and the correlated random effects one sd above/below the mean.



Clearly there are large unobserved school effects on language scores which persist at high verbal IQs.

Multilevel Models

4. Longitudinal Data. Growth Curve Models

Germán Rodríguez

Princeton University

April 4, 2018

Longitudinal data

MALMUS devotes Chapters 5-7 to models for longitudinal data with emphasis on short panels, and considers four kinds of models

1. Random-effect models, where unobserved heterogeneity at the subject level is represented by random intercepts and slopes
2. Fixed-effect models, where we introduce an additional parameter per subject to focus on within-subject variation
3. Dynamic models, where the response at a given time depends on previous or lagged responses
4. Marginal models, where focus is on population average effects and individual differences are of secondary concern

We will focus on random-effect models for longitudinal data. Many of the issues that arise here are the same as for clustered data, so we will place emphasis on aspects that are unique to panel data. We will then close with a couple of words on dynamic models.

Growth-curve models

We consider a repeated-measurements design where an outcome is measured at different times on the same individuals, leading to a *growth curve* or latent trajectory model.

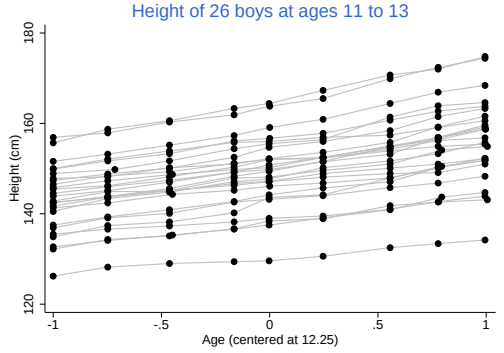
Examples include weight gain during pregnancy, or depression scores by age. The term *latent* trajectory is used because each individual follows his or her own curve over time.

Growth curve models can be fit using standard two-level models where the individual acts as the grouping level, particularly if they are extended to allow for *serial* correlation in the residuals.

If all individuals are measured at exactly the same ages, growth curves can also be modelled using structural equation models (SEM) with exactly the same results for equivalent models.

Height of boys at ages 11 to 13

We illustrate the main ideas using an example in Goldstein (1995), see §6.4 and 6.5, starting on page 91, on the heights of boys measured on nine occasions



The data are available on the course website as `oxboys.dta`, with an analysis using Stata and R at `oxboys.html`

A polynomial growth equation

The basic model used by Goldstein is a fourth-degree polynomial on age, where the constant, linear and quadratic coefficients are random at the child level, so

$$Y_{it} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})x_{it} + (\beta_2 + b_{2i})x_{it}^2 + \beta_3 x_{it}^3 + \beta_4 x_{it}^4 + e_{it}$$

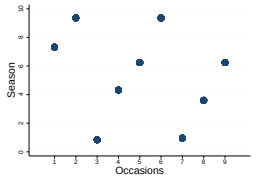
where Y_{it} is height in cm and x_{it} is age of the i-th child at time t, centered around 12 years and 3 months.

The child-level residuals (b_{0i}, b_{1i}, b_{2i}) are assumed to come from a trivariate normal distribution with mean zero and unstructured covariance matrix (with three variances and three correlations), and $e_{it} \sim N(0, \sigma_e^2)$ is the occasion-specific error term.

This is a standard random-coefficient model with the child as the grouping level, so we already know how to fit it. Let's add some bells and whistles.

Seasonality

Observations taken throughout the year may exhibit seasonality. In our dataset the boys were measured in different months of the year, as shown in a plot of `season` by `occasion`



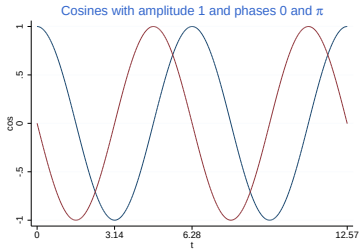
A simple model where a seasonal component has amplitude α and phase ϕ can be written as

$$\alpha \cos(t + \phi) = \alpha_1 \cos(t) - \alpha_2 \sin(t)$$

In this dataset the coefficient of the sine term was very close to zero and was omitted from the model.

Aside on cosines

For those of us who need a refresher, here's a plot of $cos(t)$ for $t \in (0, 4\pi)$ in and out of phase



To compute the cosine term we simply scale `season` to the range $(0, 2\pi)$, calculate

$$\text{sc} = \cos(\pi\, \text{seas}/6)$$

and add the resulting cosine to the fixed part of the model.

The standard model

As this point we are ready to reproduce the results in Table 6.4 in Goldstein (1995, p.93).

Please refer to the website for the code used to run the model in Stata and R. The fixed part of the model has linear, quadratic, cubic and quartic terms on age plus a seasonality term, while the random part lets the intercept and linear and quadratic age terms vary randomly across children.

How would you interpret the coefficient of the seasonality component? How much do you expect a child to grow, on average, between ages 12.25? and 13.25? What's the correlation between the heights of the same child at those two ages? Do you think the model assumptions so far are reasonable?

Serial correlation

With clustered data a random-intercept model assumes an *exchangeable* correlation structure, where any two outcomes have the same correlation, arising from the fact that they share a_i.

With longitudinal data this assumption is suspect because outcomes that are closer in time are likely to be more highly correlated than observations taken further apart.

Fortunately, we can extend the model to allow for *serially correlated* residuals. In particular, we will assume that

$$\text{cov}(e_{it_1}, e_{it_2}) = \sigma_e^2 e^{-\gamma(t_2 - t_1)}$$

which reduces to the variance σ_e^2 when $t_1 = t_2$ and decays exponentially to zero as the gap between the times increases.

Both Stata and R allows for this form of residual correlation, among others.

The full model

The computing logs show how to fit this fourth degree polynomial with seasonality, with the level, gradient and curvature by age varying across children, and residuals that are serially-correlated within each child.

Here are (somewhat abbreviated) results from Stata

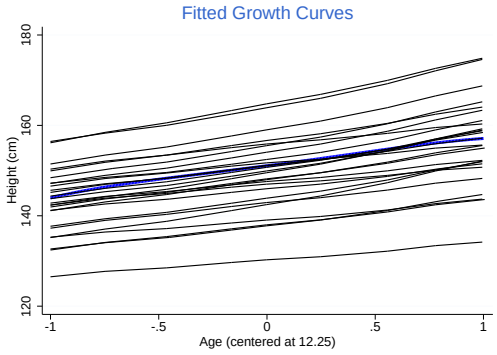
```
Wald chi2(5)    =  502.97
Log likelihood = -305.76024

------------------------------------------
      height |      Coef.   Std. Err.
-------------+----------------------------
         age |   6.190767   .3508537
        age2 |    2.16322   .4493732
        age3 |    .386329   .1690328
        age4 |  -1.548466   .4293597
          sc |  -.2360017   .0673323
       _cons |     148.911   1.539373
------------------------------------------
```

```
------------------------------------------------------
  Random-effects Parameters  |   Estimate   Std. Err.
-----------------------------+------------------------
id: Unstructured     sd(age) |    1.63716   .2346991
                    sd(age2) |   .7579632    .152763
                   sd(_cons) |   7.840658   1.088743
              corr(age,age2) |   .6869741   .1494221
             corr(age,_cons) |   .6177878   .1243386
            corr(age2,_cons) |   .2489086   .2226974
-----------------------------+------------------------
Residual: Exponential   rho |   .0010001   .0032199
                    sd(e) |    .484354   .0478213
------------------------------------------------------
```

We will examine these results largely through graphs.

Fitted grow curves

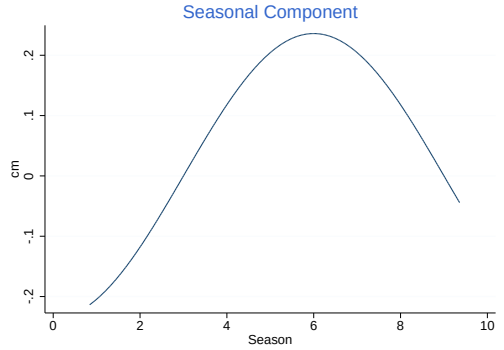
The figure shows the population average curve and the fitted growth curves for each child, using ML to estimate the fixed coefficients and EB for the random coefficients



The curves reflect substantial variation in growth curves across children, with large differences in average height.

Interpreting seasonality

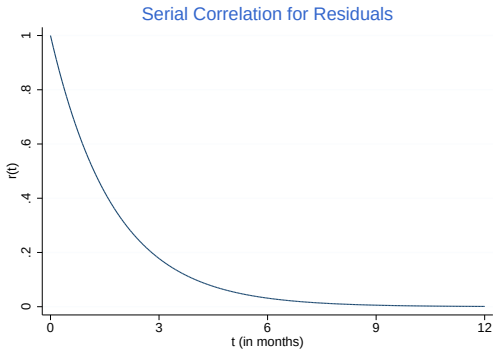
The coefficient of the cosine term or amplitude is estimated at -0.236. We can plot the estimated curve $-0.236\cos(\pi x/6)$ for $x \in (0.84, 9.36)$, the range in the data.



The estimates show that boys grow about half a centimeter more in the summer than in the winter.

Interpreting serial correlation

For residuals with a gap of t the serial correlation is $\rho(t) = e^{-\gamma t}$. Stata reports $\rho(1) = 0.001$ so $\gamma = 6.91$. We plot $\rho(t) = e^{-\gamma t}$ for t in $(0, 1)$, but label the gap in months:



The correlation between residuals is 0.178 after 3 months, and falls to 0.032 after 6 months.

Correlation among outcomes

It is important to understand that the serial correlation we have estimated is just one aspect of the correlation among outcomes in the same child, the part due to correlated residuals.

A larger part of the correlation comes from the latent trajectory, or the fact that measurements on a child on different occasions share the random intercept and slopes for the linear and quadratic terms.

In fact, the correlation between heights measured at ages 11.25 and 11.5, corresponding to the first two occasions, is estimated as 0.996 according to the model. We'll see in a minute how to obtain this result from first principles.

The observed correlation is also 0.996. The easiest way to verify this fact is to change the data to wide format.

Calculating correlations

The outcomes at ages 11.25 and 11.5 for child i involve the random effects $u_i = (a_i, b_i, c_i, e_{i1}, e_{i2})'$.

The variances and covariances of these terms can be extracted from the output and turn out to be

$$V = \begin{bmatrix} 61.476 & & & & \\ 7.930, & 2.689 & & & \\ 1.479, & 0.852, & 0.575 & & \\ 0, & 0, & 0, & 0.235 & \\ 0, & 0, & 0, & 0.042, & 0.235 \end{bmatrix}$$

The random part of the outcomes for the same child at the given ages is a linear combination of u_i with coefficients

$$C = \begin{bmatrix} 1, & -1, & 1, & 1, & 0 \\ 1, & -0.75, & 0.75^2, & 0, & 1 \end{bmatrix}$$

The variance-covariance of u_i is then CVC'.

Testing variances of random coefficients

There is no question that the curves vary by child. The table below shows reductions in deviance starting from the population average model, letting the intercept, slope and curvature be random, and finally allowing for serial correlation of residuals.

Model	log L	χ^2	*df*
Fixed coefficients	-819.79		
Random intercept	-463.62	712.33	1
Random slope	-333.26	260.73	2
Random curvature	-306.79	52.93	3
Serial correlation	-305.76	2.06	1

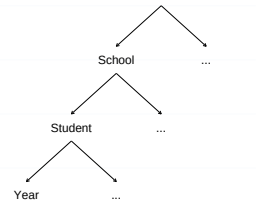
All tests are on a boundary of the parameter space and thus are conservative. All are significant except for serial correlation.

You may want to try using REML estimation to see if that makes a difference in light of the modest sample size.

Three-level Models

The computing logs have an analysis of three-level panel data with 7230 observations on 1721 students in 60 schools.

The outcome of interest is math achievement. The data were collected over six years from first to sixth grade, but not all students have six outcomes, so the panel is not balanced.



The data come from Chapter 4 in the HLM 6 manual and came in three files, which I merged into a single Stata file called `egm.dta`. The analysis may be found in `egm.html`.

A Growth Curve

The models considered in the analysis include

1. a three-level variance components model, which helps introduce intra-level correlations,
2. a growth-curve model where math scores increase linearly with year, with intercept and slopes that vary at the student and school level, and
3. a model where a student's growth curve depends on ethnicity, with different intercept and slopes for whites, blacks and hispanics, and the school average curve depends on the percent of students with low income

We follow Bryk and Raudenbush developing the models level-by-level, which helps determine which cross-level interactions to include.

Dynamic models

Consider a lagged-response model, where the outcomes at previous times are treated as covariates. For example in an autoregressive lag-1 or AR-1 model:

$$Y_{it} = \alpha + \beta x_{it} + \gamma y_{i,t-1} + e_{it}$$

where $e_{it} \sim N(0, \sigma^2)$ with independence across occasions.

This model should only be used if it makes sense to control the effect of the covariates on previous outcomes, or if the effect of the lagged response is itself of interest.

With more than two occasions some outcomes appear on both the right and left-hand sides of the equation. If the process started long before the first occasion and $\gamma < 1$ the process will be stationary.

A related approach controls for baseline conditions.

Dynamic models with random effects

The previous model is often extended by adding a random effect at the individual level to account for correlated residuals

$$Y_{it} = (\alpha + a_i) + \beta X_{it} + \gamma Y_{i,t-1} + e_{it}$$

This model poses special challenges because the lagged outcome is necessarily correlated with the random effect.

Anderson and Hsiao proposed an instrumental variables estimator using a second-order lag.

Arellano and Bond proposed a generalized method of moments estimator using additional instruments.

These approaches are both implemented in Stata, but fall beyond the scope of the course.

The Generalized Linear Mixed Model

All the multilevel models considered in this part of the course are special cases of the GLMM

$$\underset{n\times 1}{\mathbf{y}} = \underset{n\times pp\times 1}{\mathbf{X}\ \beta} + \underset{n\times qq\times 1}{\mathbf{Z}\ \mathbf{u}} + \underset{n\times 1}{\mathbf{e}}$$

where $\mathbf{X}$ is the design matrix for the fixed effects β, $\mathbf{Z}$ is the design matrix for the random effects $\mathbf{u} \sim N_q(\mathbf{0}, \mathbf{\Omega})$ and $\mathbf{e} \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$ is the error term. Usually $\mathbf{\Omega}$ is block-diagonal by level.

In this model the mean and variance are

$$E(\mathbf{y}) = \mathbf{X}\beta \quad \text{and} \quad \text{var}(\mathbf{y}) = \mathbf{Z\Omega Z}' + \sigma^2\mathbf{I}$$

Exercise: Write down the model matrices for a two-level random-intercept model with 2 observations per group.

GLMM Estimation

If the parameters in $\mathbf{\Omega}$ are known, or more generally conditional on estimates of those parameters, the maximum likelihood estimator of β can be obtained by GLS

$$\hat{\beta} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$$

Inversion of $\mathbf{V}$ takes advantage of its block diagonal structure, so the calculations are reasonably straightforward.

Using this estimator in the multivariate normal likelihood yields a profile likelihood that can then be maximized w.r.t. the parameters in $\mathbf{\Omega}$. Goldstein showed how this step can also be done using GLS.

Estimation proceeds by alternating the two steps and usually converges very quickly. Harville showed how the same steps can be adapted to use REML as proposed by Patterson and Thompson. The Longford book has details.

Multilevel Models

5. Multilevel Logit Models

Germán Rodríguez

Princeton University

April 9, 2018

Binary data

We now turn our attention to clustered and longitudinal *binary* data. Examples that we will consider include

- Data on the decision to deliver a birth in a hospital or elsewhere, with repeated observations on a sample of women.
- Contraceptive use by women in the Bangladesh DHS. The data are clustered by district, which may affect both levels and urban-rural differentials in contraceptive use.
- Immunization status for Guatemalan children, which are clustered by mother, which are in turn nested in communities.

For the first example and part of the second we can use fixed or random effects models.

For three or more levels, and more generally for random coefficient models, we need a multilevel approach.

We start with a quick reminder of fixed and random-effects models.

Fixed effects models

We consider a clustered binary outcome following the fixed-effects model

$$Y_{ij} \sim B(\pi_{ij}), \quad \text{with} \quad \text{logit}(\pi_{ij}) = \alpha_i + x'_{ij}\beta$$

where α_i is a separate parameter for each group.

The usual ML estimator, equivalent to adding a dummy variable for each group, is inconsistent not just for the group parameters α_i but for β as well, in contrast with linear models.

The solution is to condition on group totals, which happen to be minimal sufficient statistics for the group effects α_i.

The resulting likelihood involves only groups with variation in both the outcome and the predictors. Sometimes losing 90% of the data is disconcerting, but a necessary price to pay to control for group-level omitted variables.

Random effects models

An alternative model assumes that the group effects are random, so

$$Y_{ij} \sim B(\pi_{ij}), \quad \text{where} \quad \text{logit}(\pi_{ij}) = a_i + x'_{ij}\beta$$

where $a_i \sim N(0, \sigma_a^2)$, is independent of the covariates and of the implicit error term.

The model can be written in terms of a latent variable following a linear random-intercept model, where $Y_{ij} = 1$ if $Y^*_{ij} > 0$, and

$$Y^*_{ij} = a_i + x'_{ij}\beta + e_{ij}$$

where $a_i \sim N(0, \sigma_a^2)$ as before and e_{ij} has a standard logistic distribution with mean 0 and variance $\pi^2/3$ (or $N(0, 1)$ for probit). Just as in logit models we fix the error variance to identify β.

Estimation by ML is implemented in Stata and R, but is not without some challenges that we now discuss.

Maximum likelihood estimation

In multilevel linear models the marginal likelihood is multivariate normal, so estimation is straightforward.

In multilevel logit models the likelihood is logistic-normal and, unfortunately, has no closed form. In the random intercept model the contribution from cluster i is

$$L_i = \int_{-\infty}^{+\infty} g(a) \prod_{j=1}^{n_i} \pi_{ij}(a)^{y_{ij}} [1 - \pi_{ij}(a)]^{1-y_{ij}} da$$

where $\pi_{ij}(a) = \text{logit}^{-1}(a + x'_{ij}\beta)$ and $g(a)$ is the $N(0, \sigma_a^2)$ density. This integral is intractable.

Not surprisingly, various researchers have proposed approximations. Regrettably, some of them don't work very well. I'll summarize the main approaches, see Rodríguez and Goldman (1995, 2001), henceforth RG1 and RG2, for more details.

MLQ: Marginal quasi-likelihood

The multilevel logit model can be written in general form as

$$\mathbf{y} = \boldsymbol{\pi} + \mathbf{e} \quad \text{where} \quad \boldsymbol{\pi} = \text{logit}^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}u)$$

MQL-1. Goldstein approximates the inverse logit using a first order Taylor series about $\boldsymbol{\beta} = \boldsymbol{\beta}_0$ and $\mathbf{u} = \mathbf{0}$ for a trial estimate $\boldsymbol{\beta}_0$. This leads to an approximating multilevel *linear* model, which is used to obtain an improved estimate. The procedure is iterated to convergence. Longford uses a quadratic approximation to the log-likelihood. RG1 show that it is equivalent to MQL-1.

MQL-2. A second-order approximation that uses second derivatives w.r.t. $\mathbf{u}$ only, ignoring second derivatives w.r.t. the fixed effects $\boldsymbol{\beta}$ as well as mixed derivatives. Convergence can be an issue.

Both procedures are implemented in MLwiN. Not surprisingly, they work well for very small $\mathbf{u}$.

PQL: Penalized quasi-likelihood

PQL-1. An obvious improvement is to approximate $\boldsymbol{\pi}$ using a Taylor series expansion about

$$\boldsymbol{\beta} = \boldsymbol{\beta}_0 \quad \text{and} \quad \mathbf{u} = \mathbf{u}_0$$

where $\boldsymbol{\beta}_0$ is the current ML estimate of $\boldsymbol{\beta}$ and $\mathbf{u}_0$ is the EB estimate of $\mathbf{u}$ evaluated at current parameter estimates. This method has been derived by several authors using different perspectives. It was named PQL by Breslow and Clayton. It usually works better than MQL.

PQL-2. Goldstein and Rasbash proposed a second-order PQL approximation using second derivatives w.r.t. $\mathbf{u}$ only, ignoring second derivatives w.r.t. the fixed effects $\boldsymbol{\beta}$ as well as mixed derivatives, just as before.

MLwiN implements both forms of PQL.

A simulation study

RG1 conducted a simulation study using several scenarios, involving small and large random effects and designs with small and large clusters, as found in education and demographic research.

Of particular interest is a set of simulations using the same structure as a real dataset from Guatemala, which concerned prenatal care for 2449 births among 1558 women nested in 161 communities. In fact, it was doubts about conventional estimates obtained with the actual data that motivated the simulation study.

We simulated data using known values of the fixed coefficients and of the variances of the random effects, and then fitted a three-level random intercept model using MQL and PQL.

The data were made available through JRSS-A and on my website, and have been used by several authors, including Nelder, Goldstein and Rasbash, and Browne and Draper.

Comparison of estimates

Here are some results from Table 9.1 in Rodríguez (2008), which has the most complete set of estimates for a simulation using large random effects.

Estimation	Fixed Part (β)			Random Part (σ)	
method	Individual	Family	Community	Family	Community
True value	1.000	1.000	1.000	1.000	1.000
MQL-1	0.738	0.744	0.771	0.100	0.732
MQL-2	0.853	0.859	0.909	0.273	0.763
PQL-1	0.808	0.806	0.831	0.432	0.781
PQL-2	0.933	0.940	0.993	0.732	0.924

MQL-1 underestimates βs by 23-26% and σs by 27 and 90%! MQL-2 is more accurate but doesn't always converge. PQL-1 is better than MQL-1, competitive with MQL-2, and more likely to converge. PQL-2 is best in the series, with 1-7% bias for the βs, but still underestimates σs by 8 and 27% and may not converge.

Gaussian quadrature

In light of these results we turned to ML via numerical integration of the likelihood function using Gaussian quadrature.

Quadrature rules approximate an integral as a weighted sum over a grid of points. Gaussian quadrature chooses both the weights and the evaluation points to minimize error for different integrands.

Gauss-Hermite quadrature can be used with integrals of the form

$$\int f(x)e^{-x^2}dx = \sum_{k=1}^{q} w_k f(x_k)$$

The evaluation points are zeroes of the Hermite polynomials and, together with the weights, can be obtained from tables or code.

This method can be applied to the integral in slide 5 through a simple change of variables.

Adaptive quadrature

An alternative procedure that achieves remarkable accuracy with fewer points moves the evaluation points to cover the posterior rather than the prior distribution of the random effects.

Liu and Pierce approximate the posterior using a normal distribution with the same mode and curvature at the mode. This has the effect of sampling the integrand in a more relevant range. The method with just one point is equivalent to a Laplace approximation or PQL-1.

Rabe-Hesketh and collaborators, building on work by Naylor and Smith, use the posterior mean and variance of the random effects instead of the mode and curvature. This leads to somewhat simpler calculations and was first implemented in their `gllamm` command.

Pinheiro and Bates see adaptive quadrature as a deterministic version of importance sampling and use it in non-linear models.

Validating quadrature methods

Does it work? We validated ML via quadrature using the simulated data before using it on actual data, with the following results

Estimation	Fixed Part (β)			Random Part (σ)	
method	Individual	Family	Community	Family	Community
True value	1.000	1.000	1.000	1.000	1.000
ML-5	0.983	0.988	1.037	0.962	0.981
ML-20	0.983	0.990	1.039	0.973	0.979

Obviously numerical integration works very well indeed, even with as few as 5 points.

Our analysis of the Guatemalan data, published in *Demography* and used as a case study in RG2, used 20 quadrature points at each level. I later was able to reproduce the results exactly using 12-point adaptive quadrature. The page `maxlik.html` has some vintage runs and comparisons, but these days we use Stata or R.

Software notes

Stata implements quadrature procedures in two commands:

`xtlogit` fits random-intercept models. The option `intmethod()` can be `ghermite` for classic Gauss-Hermite, `aghermite` for adaptive G-H using mode and curvature, or `mvaghermite` for adaptive G-H using the mean and variance. The default is `mv`. The number of points is specified with the `intpoints()` option and defaults to 12.

`melogit` fits random-coefficient models using adaptive Gauss-Hermite with 7 points per effect as the default. In addition to the `intmethod()` and `intpoints()` options, there's a `laplace` option, equivalent to PQL-1, as a faster but less accurate alternative for exploratory work. The number of integration points can be varied by level.

Stata 14 can also fit these models using `meglm`.

Software notes (continued)

R's `lme4` package has a function `glmer()` to fit generalized linear multilevel models.

For random-intercept models the default is PQL, but it is possible to specify adaptive quadrature using the mode and curvature by specifying the number of integration points via the `nAGQ` argument, which defaults to one. I strongly recommend that you avoid the default and specify 7 or preferably 12 points as Stata does.

For random-coefficient models the only option available is PQL, which unfortunately means that maximum-likelihood results should be considered approximate and useful only for exploratory work. (As we will see later, however, these models can be estimated in R using Bayesian methods.)

Hospital Deliveries

Our first example will use data from Lillard and Panis on the decision to deliver a birth in a hospital or elsewhere, available in the datasets section as `hospital.dat`.

The dataset comprises 501 women with 1060 births. The outcome `hosp` is a binary indicator of hospital delivery with mean 0.297.

The predictors of interest are `loginc` or log-income, `distance` to the nearest hospital, and two indicators of the woman's education: `dropout` for less than high school and `college` for college graduates or higher (only 8.4% of the women).

A simple logit model suggests that all predictors have significant effects on the probability of hospital delivery, but the assumption of independence is not adequate with repeated observations on the same women.

A Random-Intercept Model

We therefore introduce a woman level random effect a_i and assume that conditional on that each woman's outcomes are independent with probability satisfying the logit model

$$\Pr\{Y_{ij} = 1|a_i\} = \text{logit}^{-1}(a_i + \boldsymbol{x}'_{ij}\boldsymbol{\beta})$$

where $\boldsymbol{x}_{ij}$ represents the predictors for the j-th birth of the i-th woman and $a_i \sim N(0, \sigma^2_a)$ is the woman-specific random effect, assumed normally distributed.

As noted earlier the likelihood for this model has no closed form and must be evaluated using numerical integration. The computing logs show results using 12 quadrature points in Stata and R. Notably R's default choice of PQL does not converge with these data, but specifying `nAGQ=7` works fine.

Maximum-Likelihood Estimates

Here are estimates obtained using all the defaults in Stata

```
Integration method: mvaghermite                 Integration pts.  =         12

                                                Wald chi2(4)      =     110.06
Log likelihood  = -522.65042                    Prob > chi2       =     0.0000
------------------------------------------------------------------------------
        hosp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      loginc |   .5622009   .0727497     7.73   0.000     .4196141    .7047876
    distance |  -.0765915   .0323473    -2.37   0.018    -.1399911    -.013192
     dropout |  -1.997753   .2556249    -7.82   0.000    -2.498769   -1.496737
     college |    1.03363   .3884851     2.66   0.008     .2722135    1.795047
       _cons |   -3.36984   .4794505    -7.03   0.000    -4.309546   -2.430134
-------------+----------------------------------------------------------------
    /lnsig2u |   .4372018   .3161192                     -.1823805    1.056784
-------------+----------------------------------------------------------------
     sigma_u |   1.244335   .1966791                      .912844    1.696203
         rho |   .3200274   .0687907                      .2020988    .4665343
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 29.61                  Prob >= chibar2 = 0.000
```

We will discuss interpretation of the fixed effects as well as the standard deviation of the random effects. (We'll leave estimation of the random effects themselves to the next example.)

Subject-Specific and Population Average

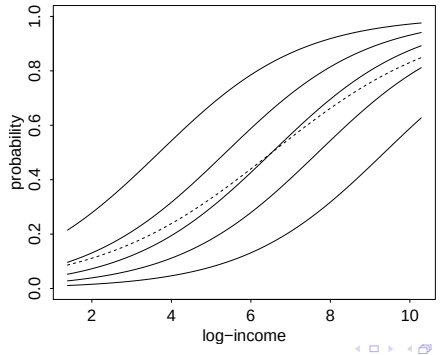
The fixed effects have a *subject-specific* interpretation. For example the coefficient of `college` means that the odds of delivering a birth in a hospital are multiplied by 2.81 when a woman has a college education, compared to what her odds would be with only a high school education but the same income, distance to the hospital, *and* unobserved characteristics as captured by a_i.

Contrast this with a *population-average* effect, which we can obtain by averaging the effect of college education over all women with given observed characteristics. For example at the mean `loginc` of 5.988 and the mean `distance` of 3.918, the probabilities for `college` 1 and 0 averaged over the distribution of a using Gauss-Hermite integration are 0.637 and 0.442, leading to an odds ratio of 2.21.

Population-average (or marginal) coefficients are smaller in magnitude than subject-specific (or conditional) coefficients.

Plotting SS and PA Effects

The figure below shows the predicted probability of hospital delivery as a function of log-income for women with high school education, who live at the average distance from a hospital, and have unobserved characteristics in percentiles 10, 30, 50, 70 and 90. We also show the predicted probabilities based on the population average model (dashed line).



Standard Deviation of Random Effects

A nice way to interpret the standard deviation σ_a is to write $a_i = \sigma_e z_i$ where z_i is a standard-normal random effect, so the model becomes

$$\text{logit}(\pi_{ij}) = \sigma_a z_i + \boldsymbol{x}'_{ij}\boldsymbol{\beta}$$

and σ_a can be interpreted as a regular logit coefficient for the standardized random intercept z_i

In our data $\hat{\sigma}_e = 1.244$. Thus, the odds of hospital delivery for a woman with unobserved characteristics one standard deviation above the mean are 3.47 times the odds of an average woman with the same log-income, distance to a hospital and education.

Similarly, the odds for a woman with unobserved characteristics one standard deviation below the mean are 71% lower than for the average woman with the same observed characteristics.

This parameter is also related to the intra-class correlation.

Latent Intra-Class Correlation

The intraclass correlation is best defined in terms of the latent variable formulation of the model shown earlier. For a logit model

$$\rho = \frac{\sigma_a^2}{\sigma_a^2 + \pi^2/3}$$

because in a standard logistic distribution $\sigma_e^2 = \pi^2/3$. (In a probit model $\sigma_e^2 = 1$, and in a c-log-log model is it $\sigma_e^2 = \pi^2/6$.)

For the hospital delivery data the correlation between the propensity of a woman to deliver any two births in a hospital is

$$\hat{\rho} = \frac{1.244^2}{1.244^2 + \pi^2/3} = 0.32$$

This also means that 32% of the variance in the latent propensity to deliver a birth in a hospital can be attributed to women.

Manifest Intra-Class Correlation

In a 2003 paper with Elo we proposed looking at the correlation between actual binary outcomes, which depends on the covariates. Our method is described in MALMUS §10.9.3 and implemented in a Stata command called `xtrho`.

We calculate a two-by-two table of expected outcomes for two observations in the same group, which we do by integrating out the random effect at selected values of the linear predictor. At the median we get

	No	Yes	
No	0.6153	0.1454	0.7607
Yes	0.1454	0.0938	0.2393
	0.7607	0.2393	1.0000

The marginal probability that a median woman would deliver a birth in a hospital is 24%, and the joint probability for two births is 9%. The Pearson correlation is 0.20 and Yule's Q is 0.46. The odds ratio is 2.73.

Correcting Standard Errors for Clustering

Some researchers faced with repeated binary observations simply fit logit models and then adjust the standard errors for clustering using extensions of the Huber-White "sandwich" estimator. This approach is fine if you keep in mind two caveats:

1. You must realize you are fitting a population-average rather than a subject-specific model and interpret the parameters accordingly. As we have seen, the effect for a particular subject differs from the average effect in the population.
2. The estimates obtained using a logit model are not efficient because they ignore the correlation structure of the data. A better approach is to use generalized estimating equations (GEE), which produces efficient population-average estimates and correct standard errors.

Comparison of Estimates

The table below compares four estimates of the effect of college education and its standard error, using logit models, logit with corrected standard errors, GEE, and random effects

	Logit	Cluster	GEE	Multilevel
$\hat{\beta}$	0.8217	0.8217	0.8078	1.0336
s.e.	0.2611	0.2884	0.2980	0.3885

The first three methods estimate a population-average effect equivalent to an odds ratio of 2.24 (not unlike our result), and both correcting for clustering and using GEE inflate the standard error.

The estimated subject-specific effect corresponds to an odds ratio of 2.81 and is larger than the average effect (it also has a larger standard error).

The key point is that having clustered data affects not just the standard errors but the coefficients themselves.

Multilevel Models

6. Multilevel Logit Models (continued)

Germán Rodríguez

Princeton University

April 11, 2018

Contraceptive Use in Bangladesh

Our second dataset concerns contraceptive use in Bangladesh from Huq and Cleland (1990) and makes an appearance in the Stata manual, Bates's `lme4` book, and other papers.

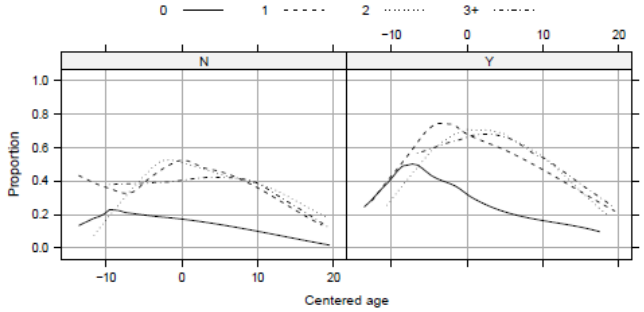
The data pertain to 1934 women grouped in 60 districts. The outcome is a binary indicator of current contraceptive use. The predictors of interest include age and number of children, as well as an indicator of urban residence. We also have a district identifier.

Most districts have both urban and rural parts. We will entertain random-intercept models where the level of contraceptive use varies by district, and random-slope models where the urban-rural differential varies by district, in both cases net of observed covariates.

The first issue, however, is how to specify the fixed part of the model.

Plotting the Data

Plotting binary data is harder than continuous data, but still necessary. A useful tool is to use scatterplot smoothers such as splines or loess. The figure below shows contraceptive use by age for rural and urban women grouped by number of children.



Contraceptive use is clearly a non-linear function of age, but many analyses use just a linear term. Kudos to Bates!

Model Selection

Perhaps the first model to fit is a random-intercept model with a linear term on age, indicators of 1, 2 and 3 or more children, and an indicator for urban residence, the model used in the Stata manual and in several previous analyses.

Following Bates we'll introduce a quadratic term on age. This addition improves the fit by a remarkable 44.12 points in the chi-squared scale, which is not surprising in light of the graph.

The figure also suggests that there are very small differences between 1, 2 and 3+ children, so we'll follow Bates and use a single indicator for any children, losing 0.37 χ^2 points while saving 2 d.f.

A final improvement is to add an interaction between the linear term on age and the indicator for children. This allows the curve for mothers to have a different peak and improves the fit by 8.00 chi-squared points at the expense of one d.f.

The Random-Intercept Model

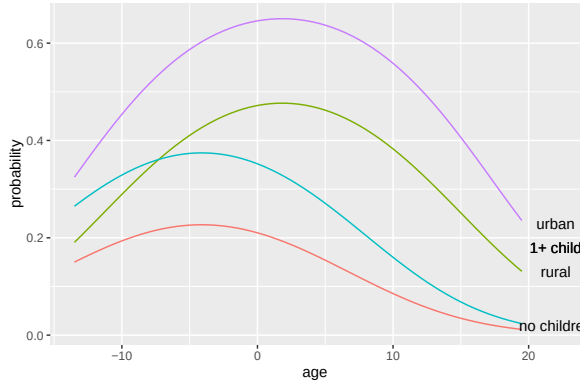
Here's the Stata output from the final random-intercept model

```
Integration method: mvaghermite                 Integration pts.  =        12

                                                Wald chi2(5)      =    146.77
Log likelihood  = -1182.4584                    Prob > chi2       =    0.0000

------------------------------------------------------------------------------
       c_use |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       urban |   .7134563   .1213548     5.88   0.000     .4756053    .9513074
         age |  -.0472872    .021841    -2.17   0.030    -.0900949   -.0044795
        age2 |  -.0057577   .0008414    -6.84   0.000    -.0074068   -.0041086
       child |   1.210876   .2075937     5.83   0.000     .8039994    1.617752
  ageXchild |   .0683467   .0254687     2.68   0.007     .0184289    .1182645
       _cons |  -1.323606   .2154606    -6.14   0.000    -1.745901   -.9013106
-------------+----------------------------------------------------------------
    /lnsig2u |   -1.48611   .3397138                     -2.151937   -.8202836
-------------+----------------------------------------------------------------
     sigma_u |   .4756585   .0807939                      .3409673    .6635561
         rho |   .0643468   .0204529                      .0341322    .1180392
------------------------------------------------------------------------------
LR test of rho=0: chibar2(01) = 44.46                  Prob >= chibar2 = 0.000
```

Results using R's `glmr()` are very similar. Try your hand at interpreting these results before peaking at the next slide.

Use by Age, Children and Residence

Seems clear that contraceptive use increases with age and then declines, is higher for women with children than those without, peaks at a later age for women with children, and is generally higher in urban areas. These effects are best shown in a graph



Variation in Use Across Districts

There is also evidence of substantial variation in contraceptive use across districts:

- The estimated standard deviation of the intercept, 0.476, means that the odds of using contraception are 60% higher in a district one standard deviation above the mean than in an average district, everything else being equal.
- The intraclass correlation between the latent propensity to use contraception of two women in the same district is 0.06. Equivalently, we can say that districts account for only 6% of the variation in propensity to use contraception net of the observed covariates.

In case you are curious the manifest correlation at the median, calculated using `xtrho`, is equivalent to an odds ratio of 1.23.

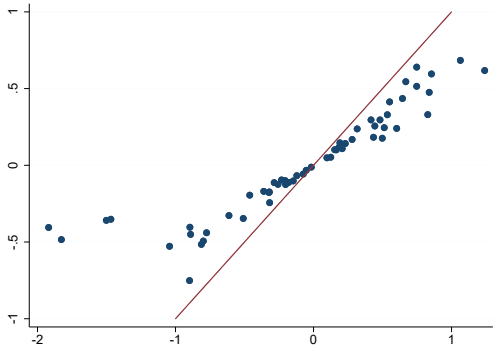
Estimation of the Random Effects

We can identify districts where women are more or less likely to use contraception by predicting the random effects. There are two ways to proceed:

1. Calculate maximum likelihood estimates by treating the estimated linear predictor from the multilevel model as an offset and then running a separate logit model in each district. The estimate is not defined when all women in a district have the same outcome, which happens in three districts.
2. Compute empirical Bayes estimates using the mean or mode of the posterior distribution of the random effects, which requires using numerical integration.

Comparison of ML and EB estimates

The graph below compares EB and ML estimates and shows the usual shrinkage towards zero.



The shrinkage is particularly noticeable in four districts, all with fewer than 15 women and effects quite far from zero.

Predicted Probabilities

Subject-specific probabilities are easily computed from first principles by setting the observed covariates and the random effects to selected values. The predicted probabilities for women of average age with children in urban and rural areas of the average district are 0.6458 and 0.4718, an odds-ratio of 2.04.

Population-average probabilities can also be computed, although they require integration over the distribution of the random effect, which can be done "by hand" or using `gllamm`. Using 12-point quadrature we obtain population-average probabilities of 0.6389 and 0.4732, or an odds-ratio of 1.97

As usual the population average effect is smaller than the subject-specific, but the difference here is modest because the intra-class correlation is low.

The Random-Slope Model

The next step is to see whether the urban-rural differential in contraceptive use varies by district, which we'll do by treating the urban effect as a random slope.

This model is analogous to allowing an interaction between urban residence and district, but instead of estimating a separate urban-rural difference for each district, we assume that they are drawn from a normal distribution. Estimation is possible because most districts have urban and rural areas; in fact we find only 15 districts with no rural women and 3 with no urban women.

The Random-Slope Model

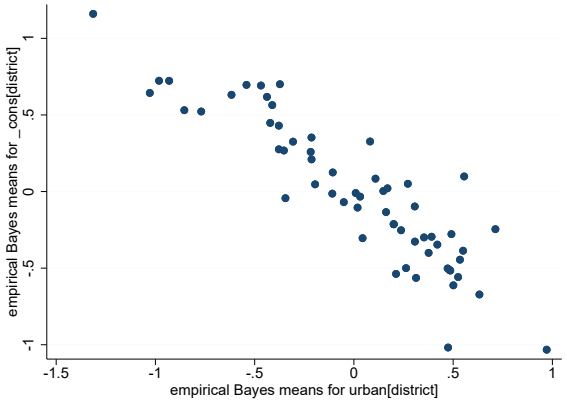
Here's the output from Stata using 7 points per effect

```
  Integration method: mvaghermite                 Integration pts.  =         7

                                                  Wald chi2(5)      =    135.42
Log likelihood = -1176.4767                       Prob > chi2       =    0.0000
------------------------------------------------------------------------------
           c_use |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-----------------+------------------------------------------------------------
           urban |   .7906825   .1648731     4.80   0.000     .4675372    1.113828
             age |  -.0461515   .0219589    -2.10   0.036    -.0891903   -.0031128
           agesq |  -.0056484   .0008514    -6.63   0.000    -.0073171   -.0039797
           child |   1.211711   .2091521     5.79   0.000     .8017802    1.621641
      ageXchild |    .066423   .0256306     2.59   0.010     .0161879     .116658
           _cons |  -1.344866   .2244245    -5.99   0.000     -1.78473   -.9050024
-----------------+------------------------------------------------------------
district         |
      var(urban)|   .5453645   .2931897                      .1901421    1.564212
      var(_cons)|   .3859845   .1280172                      .2014915    .7394059
-----------------+------------------------------------------------------------
district         |
 cov(urban,_cons)|   -.363198   .1660099    -2.19   0.029    -.6885714   -.0378246
------------------------------------------------------------------------------
LR test vs. logistic model: chi2(3) = 56.42              Prob > chi2 = 0.0000
```

The negative covariance should reinforce the importance of specifying `covariance(unstructured)`.

Empirical Bayes Estimates

Let us look at estimates of district effects on rural levels and urban-rural differentials in contraceptive use. We could compute ML estimates as we did for the random intercept model, but I will focus on EB estimates



Empirical Bayes Estimates (continued)

We see a clear negative correlation as noted earlier. Districts where contraceptive use in rural areas is higher than expected after considering the age and motherhood status of women, tend to have a smaller urban-rural differential in contraceptive use.

An alternative parametrization estimates separate urban and rural levels and omits the constant in the fixed and random parts. This formulation leads to exactly equivalent estimates of the fixed part but the two random effects turn out to be nearly independent. Details are left as an exercise.

Immunization in Guatemala

The final example is our analysis of childhood immunization in Guatemala. This is a three-level dataset with 2159 children of 1595 mothers who live in 161 communities, analyzed in our *Demography* paper and RG2, and used as a detailed illustration of 3-level models in MALMUS §16.2-16.8, pages 873–897.

The sample consists of children age 1-4 who have received at least one immunization, and the outcome of interest is whether they have received the full set appropriate for their age. Predictors include

1. age of child at child level,
2. mother's ethnicity and education and father's education at the family level, and
3. urban indicator and percent indigenous in 1981 at the community level.

We will return to this dataset when we compare those results with Bayesian methods.

Multilevel Models

7. Bayesian Inference in GLMMs

Germán Rodríguez

Princeton University

April 16, 2018

Generalized linear multilevel models

The logit models we have discussed are special cases of the generalized linear mixed/multilevel model. We assume that conditional on a set of multivariate normal random effects

$$\mathbf{u} \sim N_q(\mathbf{0}, \mathbf{\Omega})$$

the outcome $\mathbf{y}$ has a distribution in the *exponential family*, which includes the normal, binomial, Poisson, gamma and others. We further assume that the conditional expectation satisfies

$$E(\mathbf{y}|\mathbf{u}) = f^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u})$$

where $\mathbf{X}$ is the model matrix for the fixed effects $\boldsymbol{\beta}$, $\mathbf{Z}$ is the model matrix for the random effects $\mathbf{u}$, and $f()$ is a one-to-one transformation called the *link* function, which includes the identity, logit, probit, c-log-log, log, and others.

The marginal likelihood has a closed form for normal models with identity link, but otherwise involves intractable integrals.

Maximum likelihood estimation

ML estimates can be computed by numerical integration of the likelihood function using Gaussian quadrature, but the procedure is computationally intensive and can only be used for simple models.

A two-level random-intercept logit or poisson model requires a one-dimensional integral. Using 12 quadrature points is equivalent to 12 logit or Poisson likelihoods.

A three-level random intercept model, or a two-level model with a random intercept and slope, requires a two-dimensional integral. One evaluation of a logit or Poisson likelihood using 12 quadrature points per level is equivalent to 144 one-level models.

A three-level logit model with two random coefficients per level using 12-point quadrature for each, is equivalent to evaluating almost 21,000 logit or Poisson likelihoods. Numerical integration doesn't scale well, and soon succumbs to the "curse of dimensionality".

Bayesian estimation

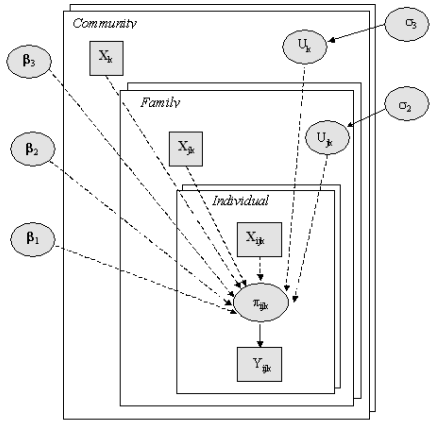
Recent advances in Bayesian estimation avoid the need for numerical integration by taking repeated samples from the posterior distribution of the parameters of interest.

To apply this framework we adopt a Bayesian perspective, treating all parameters as random variables and assigning prior (or hyperprior) distributions to the fixed parameters $\boldsymbol{\beta}$ and to the variances $\mathbf{\Omega}$ of the random effects. (We have, of course, already assigned a prior distribution to the random effects $\mathbf{u}$.)

To obtain Bayesian estimates that are roughly comparable to maximum likelihood estimates, many analysts use vague or non-informative priors. Fixed effects are typically assumed to come from normal distributions with mean zero and very large variances. Precisions, defined as the reciprocals of variances, are often sampled from diffuse gamma distributions. Gelman suggests using a uniform prior on the standard deviation instead.

A graphical model

Bayesian models are often shown in graphical form as illustrated below for a 3-level random intercept logit model



Here we need priors for the βs and hyperpriors for the σs.

The Gibbs sampler

A popular method for drawing observations from a posterior is the Gibbs sampler, which draws from a joint distribution by sampling repeatedly from each of the full conditional distributions.
In our case we need to sample from the joint distribution of the parameters given the data, which we write as

$$[\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \mathbf{u}|\mathbf{y}]$$

The Gibbs sampler tells us that we can sample instead from the three full-conditionals

$$[\boldsymbol{\beta}|\boldsymbol{\sigma}^2, \mathbf{u}, \mathbf{y}], \quad [\boldsymbol{\sigma}^2|\boldsymbol{\beta}, \mathbf{u}, \mathbf{y}], \quad \text{and} \quad [\mathbf{u}|\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \mathbf{y}]$$

which in our case further simplify to

$$[\boldsymbol{\beta}|\mathbf{u}, \mathbf{y}], \quad [\boldsymbol{\sigma}^2|\mathbf{u}] \quad \text{and} \quad [\mathbf{u}|\boldsymbol{\beta}, \boldsymbol{\sigma}^2, \mathbf{y}]$$

The fixed effects depend only on the random effects and response, and the variances depend only on the random effects.

Markov chains

Let $\boldsymbol{\beta}_k, \boldsymbol{\sigma}^2_k$, and $\mathbf{u}_k$ denote a sample, with $k = 0$ representing initial values of the fixed and random parameters. The Gibbs sampler draws

$$\begin{array}{lll} \boldsymbol{\beta}_{k+1} & \text{from} & [\boldsymbol{\beta}|\mathbf{u}_k, \mathbf{y}] \\ \boldsymbol{\sigma}^2_{k+1} & \text{from} & [\boldsymbol{\sigma}^2|\mathbf{u}_k] \quad \text{and} \\ \mathbf{u}_{k+1} & \text{from} & [\mathbf{u}|\boldsymbol{\beta}_{k+1}, \boldsymbol{\sigma}^2_{k+1}, \mathbf{y}] \end{array}$$

This is a Markov chain because each sample depends only on the previous one. Under reasonably general conditions, the sample converges in distribution to the joint posterior of interest.

Usually one discards a "burn-in" period long enough to ensure that the chain has converged to its stationary distribution and uses the remaining observations to estimate features of the posterior, such as the mean or a credible interval.

The sample is **not** i.i.d. The efficiency of the chain is lower when the draws are highly correlated.

Sampling methods

The actual sampling is done using methods appropriate for each distribution. I'll mention just a couple of approaches.

Uniform Distribution. An indispensable starting point is a routine to generate pseudo-random numbers or samples from the uniform distribution in $(0, 1)$, which both Stata and R do well. `runiform` in Stata.

The Inversion Method. A useful general method is based on the fact that

$$\text{if } X \sim F(x) \quad \text{then} \quad F(X) \sim U(0, 1)$$

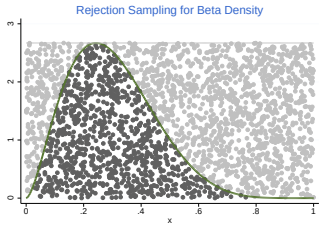
If we can invert the c.d.f. we can then draw samples from it by calculating $F^{-1}(u)$ where $u \sim U(0, 1)$.

For example we could draw normals this way. But Stata and R have specialized function for many distributions including beta, binomial, χ^2, gamma, hypergeometric, normal, Poisson, and

Rejection sampling

What if the distribution you need, say $f(x)$, isn't in the list? There's an ingenious method called rejection (or importance) sampling which has wide applicability.

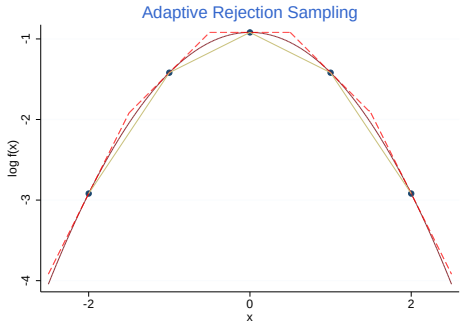
All you need is another density that you know how to sample from, say $g(x)$, which "covers" $f(x)$ in the sense that it has the same domain and there is a constant c such that $cg(x) \geq f(x)$ for all x.



You then draw a sample x from $g(x)$ and keep it with probability $f/cg(x)$, which you do by sampling u from $U(0,1)$ and comparing it to the ratio above. This corrects for the fact that sampling from $g(x)$ oversamples values of x where $g(x)$ is "taller" than $f(x)$.

Adaptive rejection sampling (ARS)

Gilks proposed a sampling method that extends rejection sampling to any log-concave distribution, using outer and inner envelopes based on $f(x)$ and its derivative $f'(x)$ at selected points.



The method samples from the outer envelope, accepts values under the inner envelope, and otherwise evaluates $f(x)$ to decide and then $f'(x)$ to tighten the envelopes.

WinBUGS

Gibbs sampling using adaptive rejection sampling has been implemented in a package called BUGS (Bayesian Inference Using the Gibbs Sampler). The windows version is called WinBUGS.

Two alternatives are OpenBUGS and JAGS (Just Another Gibbs Sampler), as well as MLwIN, but we'll focus on WinBUGS. The program lets you describe your model using a declarative language to specify the prior distributions and the likelihood, and uses an expert system to derive the posterior and decide whether to use a specialized sampling method or ARS. We'll consider two examples:

1. An analysis of immunization in Guatemala based on a three-level random-intercept model as illustrated in slide 5, comparing results with other methods.
2. A study of hospital delivery data from Lillard and Panis.

In the process we will address the important issue of convergence diagnostics.

Immunization in Guatemala

Here are results from the immunization model described in RG2

TABLE 2. Estimates for Multilevel Model of Complete Immunization Among Children Receiving Any Immunization

	Logit	MQL-1	MQL-2	PQL-1	PQL-2	PQL-B	ML	Gibbs
Fixed Effects								
Individual								
*Child age 2+	0.95	0.93	1.11	0.98	1.44	1.80	1.72	1.84
Mother age 25+	-0.08	-0.08	-0.10	-0.09	-0.16	-0.19	-0.21	-0.26
Birth order 2-3	-0.08	-0.09	-0.11	-0.10	-0.19	-0.15	-0.26	-0.29
Birth order 4-6	0.09	0.13	0.15	0.13	0.17	0.27	0.18	0.21
Birth order 7+	0.15	0.19	0.23	0.20	0.33	0.39	0.43	0.50
Family								
Indigenous no Spanish	0.28	-0.04	-0.05	-0.05	-0.13	-0.06	-0.18	-0.22
Indigenous Spanish	0.22	0.01	0.01	0.00	-0.05	0.03	-0.08	-0.11
Mother educ primary	0.25	0.21	0.25	0.22	0.34	0.42	0.43	0.48
Mother educ sec+	0.30	0.22	0.27	0.23	0.34	0.46	0.42	0.46
*Husband educ primary	0.29	0.28	0.34	0.30	0.44	0.57	0.54	0.59
Husband educ sec+	0.21	0.25	0.31	0.27	0.41	0.47	0.51	0.55
Husband educ missing	0.03	0.02	0.02	0.02	0.01	0.07	-0.01	0.00
Mother ever worked	0.25	0.19	0.24	0.20	0.31	0.37	0.39	0.42
Community								
*Rural	-0.50	-0.47	-0.57	-0.50	-0.73	-0.93	-0.89	-0.96
*Prop. Indigenous 1981	-0.78	-0.64	-0.78	-0.67	-0.95	-1.21	-1.15	-1.22
Random Effects								
Standard Deviations (σ)								
Family	-	0.63	0.72	0.73	1.75	2.69	2.32	2.60
Community	-	0.53	0.55	0.56	0.84	1.06	1.02	1.13
Intraclass Correlations (ρ)								
Family	-	0.17	0.20	0.20	0.53	0.72	0.66	0.71
Community	-	0.07	0.07	0.07	0.10	0.10	0.11	0.11

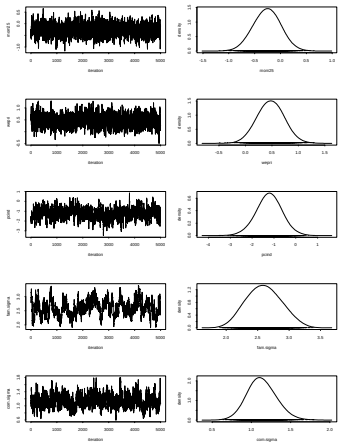
Note: asterisks indicate fixed effects significant at the five percent level according to the maximum likelihood analysis. The reference categories are child age one, mother's age < 25, birth order one, ladino, mother no education, husband no education, mother never worked and urban residence.

Trace plots and posteriors

Here is what the actual output looks like for selected parameters.

The first three parameters are child, mother and community fixed effects, the last two are standard deviations of mother and community random effects.

On the left we see trace plots, which ideally should look like fuzzy caterpillars, but the family σ shows slow mixing. On the right we see kernel estimates of the posterior densities.



Convergence diagnostics

Some Bayesians recommend using several chains and others prefer one long chain. A good compromise is to run three chains with different starting points.

For the Guatemala data we ran burn-ins of 200 followed by 5,000 draws. A battery of tests showed that this was adequate, based on

Geweke (1992)'s test of convergence, which divides the chain into two sections (such as first 10% and last 50%) and compares means

Raftery and Lewis (1992, 1996) `gibbsit` software, to determine the sample size needed to estimate each posterior c.d.f. at 95% credible limits within 0.015 with probability 0.95

Roberts (1996) estimate of efficiency, to ensure that we had the equivalent of at least 100 i.i.d. observations in the worst case, where efficiency was only 2%.

Convergence diagnostics

There's a specialized package for convergence diagnostics and output analysis for Gibbs output called CODA. The ecology is richer in the R world than in Stata, but see Thompson, Palmer and Moreno (2006) in the Stata Journal 6:530-549, available at http://www.stata-journal.com/sjpdf.html?articlenum=st0115

The website illustrates the use of winBUGS to estimate a logit model for the hospital delivery data. This model is simple enough that it can be fitted by maximum likelihood using Stata or R, but it is instructive to try the Bayesian approach, which gives similar results if we use non-informative priors.

The pages at `hospBUGS.html` and `hospBUGS2.html` have step-by-step instructions for running the model using the GUI with a compound document, and using the scripting facility introduced with version 1.4. To get winBUGS visit `http://www.mrc-bsu.cam.ac.uk/software/bugs/the-bugs-project-winbugs/`.

Multilevel Models

8. Bayesian Inference via Metropolis-Hastings

Germán Rodríguez

Princeton University

April 18, 2018

Markov Chain Monte Carlo

The Gibbs sampler is very popular but by no means the only MCMC method. An alternative is the Metropolis-Hastings algorithm, which can sample from a multivariate distribution in one step. Robert and Casella (2010) have a nice introduction to Monte Carlo Methods with R.

The basic idea given a target density f is to build a Markov Chain that has stationary distribution f. You'd think this is hard, but in fact there are methods that work in principle for any density. One such method is Metropolis-Hastings. (Another is Gibbs sampling.)

Stata now has Bayesian methods using Metropolis-Hastings, and R has an interface to Stan, which implements a variant of the algorithm using Hamiltonian dynamics, and includes a package that can fit many standard models by calling Stan to do the work.

We describe some basic features of the algorithm before turning to the implementations.

Metropolis Hastings

For a target f we need a density $q(y|x)$ that is easy to sample from (for example multivariate normal) and such that the ratio $f(y)/q(y|x)$ is known up to a constant independent of x.

If $q(.|x)$ has enough variation to cover the support of f we can build a chain that has stationary distribution f using a surprisingly simple algorithm:

Given $x^{(t)}$,

1. Generate $Y_t \sim q(y|x^{(t)})$, and
2. Take $x^{(t+1)} = Y_t$ with probability $\rho(x^{(t)}, Y_t)$ and $x^{(t)}$ otherwise, where

$$\rho(x, y) = min\{\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}, 1\}$$

is the *acceptance probability*. Note that sometimes we keep the old value! The kernel q is called the *proposal* and affects the acceptance rate and efficiency of the chain.

Independent and Random Walk Variants

The basic algorithm allows the draw to depend on the current state of the chain, but this is not necessary and the proposal can be $q(y|x) = q(y)$. This leads to a simplified algorithm called *independent MH*, which is simple but hard to tune well.

One way to take into account the previous value is to simulate $Y_t = X^{(t)} + \epsilon_t$ where ϵ_t is a random perturbation with distribution g independent of $X^{(t)}$, so the proposal density $q(y|x)$ has the form $g(y - x)$, leading to the *random walk MH*

Given $x^{(t)}$,

1. Generate $Y_t \sim g(y - x^{(t)})$, and
2. Take $x^{(t+1)} = Y_t$ with probability $\min\{f(Y_t)/f(x^{(t)}), 1\}$ and $x^{(t)}$ otherwise

In fact this was the original version of the algorithm. Sometimes, howevever, random walks are slow to converge, and efficiency is highly dependent on the choice of g.

Hybrid or Hamiltonian Monte Carlo (HMC)

The latest development in MCMC is a hybrid algorithm that uses Hamiltonian dynamics borrowed from physics to improve on traditional Metropolis-Hastings by producing proposals far from the current values yet with high probability of acceptance.

The Hamiltonian of a system describes the movement of a particle given its position and momentum in space and leads to differential equations for its trajectory over time.

In statistical MCMC we treat minus the log of the posterior density as the position, and sample the momentum along each dimension from independent Gaussian distributions.

The trajectory is simulated in discrete time using L steps of size ϵ using a method known as *leapfrog* to reach a proposed state, which is then accepted or rejected using H-M with appropriate acceptance probability. See Neal (2011) for an excellent discussion.

The No-U-Turn Sampler (NUTS)

One difficulty with HMC is that it needs the gradient of the log posterior in order to compute momentum. But this can be handled using automatic differentiation.

Another difficulty is the need to fine tune the two HMC parameters L and ϵ, which is essential to obtain an efficient algorithm.

Hoffman and Gelman (2014) proposed an HMC variant known as NUTS that avoids the need to specify the number of steps L while ensuring that the trajectory is followed long enough, and can auto-tune ϵ using a clever scheme to achieve the same efficiency as HMC, and sometimes even better.

The end result is an algorithm that seems very well suited for automatic Bayesian inference without the need for costly tuning steps or substantial expertise.

Stan

The NUTS variant of the HMC algorithm has been implemented in the program Stan, a "probabilistic programming language" from Gelman's group, named after Stanislaw Ulam, inventor of Monte Carlo. The language has a website at http://mc-stan.org.

Stan is a high-level language not unlike BUGS that can be used to specify a model, but then generates a C++ program that is compiled and run to generate the samples efficiently.

There are interfaces to run Stan from R and Stata (as well as Python, Julia, Matlab, Mathematica and Scala) which help a bit, but still require learning the modeling language.

There is also an R package called RStanArm that makes using Stan extremely easy for standard models because you can specify them using R syntax!

The Hospital Data

My first experience with Stan was running the Lillard and Panis hospital delivery data. Here's the code, saved in R as a string:

```
data {
    int N; // number of obs (pregnancies)
    int M; // number of groups (women)
    int K; // number of predictors

    int y[N]; // outcome
    row_vector[K] x[N]; // predictors
    int g[N];    // map obs to groups (pregnancies to women)
}
parameters {
    real alpha;
    real a[M];
    vector[K] beta;
    real<lower=0,upper=10> sigma;
}
model {
  alpha ~ normal(0,100);
  a ~ normal(0,sigma);
  beta ~ normal(0,100);
  for(n in 1:N) {
    y[n] ~ bernoulli(inv_logit( alpha + a[g[n]] + x[n]*beta));
  }
}
```

The variable names are not very descriptive because I wanted to write code I could use for other random-intercept logit models.

Running from R

To run the model I first copied the data from `hosp` to a list with the same names as the Stan code

```
hosp_data <- list(N=nrow(hosp),M=501,K=4,y=hosp[,1],x=hosp[,2:5],g=hosp[,6])
```

I then ran the model specifying 2 chains of 2000 samples each

```
hfit <- stan(model_code=hosp_code, model_name="hospitals", data=hosp_data, iter=2000, chains=2)

TRANSLATING MODEL 'hospitals' FROM Stan CODE TO C++ CODE NOW.
COMPILING THE C++ CODE FOR MODEL 'hospitals' NOW.
...
SAMPLING FOR MODEL 'hospitals' NOW (CHAIN 1).
Iteration: 2000 / 2000 [100%]  (Sampling)
Elapsed Time: 58.065 seconds (Warm-up)
              24.373 seconds (Sampling)
              82.438 seconds (Total)

SAMPLING FOR MODEL 'hospitals' NOW (CHAIN 2).
Iteration: 2000 / 2000 [100%]  (Sampling)
Elapsed Time: 58.074 seconds (Warm-up)
              23.186 seconds (Sampling)
              81.26 seconds (Total)
```

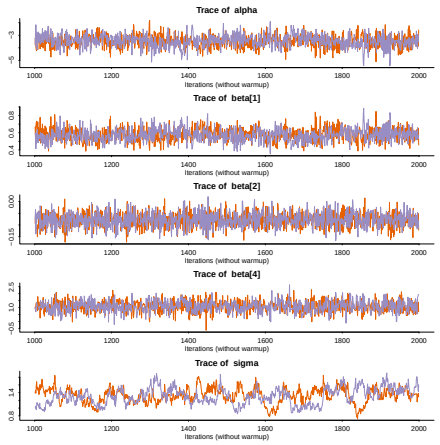
The computing log at `hospStan.html` has more details about this run.

Stan and hospital deliveries

Here are the Stan trace plots for the parameters, showing two chains.

For generality I used a vector of coefficients β. The actual names are `loginc`, `distance`, `dropout` and `college`.

As you can see, we get mostly fuzzy caterpillars, but the standard deviation of the random effects at the woman level exhibits slow mixing.



Stan Meets Applied Regression Modeling

The R package `RStanArm` makes it very easy to run the types of models in Gelman and Hill's ARM book by providing an R interface almost identical to `glm` and `glmer` to specify the model, which is then run in Stan using pre-compiled code.

Here's the R call for maximum likelihood:

```
glmer(hosp ~ loginc + distance + dropout + college + (1 | mother),
    data = hosp, family = binomial, nAGQ = 12)
```

And here's the equivalent R call for Bayesian estimation:

```
stan_glmer(hosp ~ loginc + distance + dropout + college + (1 | mother),
    data = hosp, family = binomial)
```

This will run four chains with burn-ins and samples of 1,000 observations each.

I recommend using this interface for standard models and then learning the more powerful Stan language to fit a much wider variety of realistically complex models.

Metropolis-Hastings in Stata

Stata now has a `bayeshm` command that can fit a variety of models using a random walk Metropolis-Hastings algorithm. The developers note that the algorithm is not optimal for Bayesian multilevel models, but can be used in models that do not have too many random effects. Here's a command that will run a random-intercept model with the hospital data

```
bayesmh hospital loginc distance dropout college ibn.group ///
  , likelihood(logit) ///
  prior({hospital:i.group}, normal(0,{var})) ///
  prior({hospital:loginc distance dropout college _cons}, normal(0,1000)) ///
  prior({var}, igamma(0.001,0.001)) ///
  block({hospital:i.group}, reffects) ///
  block({hospital:loginc distance dropout college _cons}) ///
  block({var})
```

The syntax is similar to other Stata commands, treating the grouping variable as a factor without a reference cell. Sampling all parameters together is inefficient and we work in blocks, separating the fixed, random and variance parameters. See the computing log at `hospStata.html` for details.

Multilevel Models

9. Models for Count and Survival Data

Germán Rodríguez

Princeton University

April 23, 2018

Poisson Models

This unit concerns models for count data. We assume that conditional on unobserved random effects the outcomes have a Poisson distribution.

For example in a two-level random intercept model we write

$$Y_{ij}|a_i \sim P(\mu_{ij}) \quad \text{where} \quad \log \mu_{ij} = (\alpha + a_i) + x'_{ij}\beta$$

We will assume that $a_{ij} \sim N(0, \sigma^2_a)$ as we have done for other models. This choice generalizes to more general random-coefficient models but requires quadrature. Stata uses adaptive quadrature in `xtpoisson` and `mepoisson` and R's `glmer()` uses quadrature for one random effect and PQL otherwise.

An alternative with Poisson models is to use a gamma-distributed multiplicative random effect, which can be integrated analytically, but doesn't generalize to correlated random effects. Stata's `xtpoisson` implements gamma as an option.

A Random-Intercept Poisson Model

Our first application is to small area estimation using data on lip cancer from Scotland. The data consist of the number of cases observed in each of 56 counties in 1975-80, and are available at http://www.stata-press.com/data/mlmus3/lips.dta.

We also have information on the expected number of cases based on age-specific lip cancer rates for the whole of Scotland and the age distribution in each county. The ratio of observed to expected counts, usually times 100, is called the Standardized Mortality Ratio (SMR). For example a value of 193.2 denotes almost twice as many cases as expected.

A limitation of crude SMRs is that estimates for counties with small populations are very imprecise. To address this problem we will use Empirical Bayes (EB) estimates based on a random-intercept Poisson model. By adding a random effect at level one we are effectively modeling over-dispersion.

Fitting the Random-Intercept Model

In this model the conditional distribution of the count is Poisson with mean proportional to the expected number of cases

$$Yi|a_i \sim P(\mu_i) \quad \text{with} \quad \log \mu_i = \alpha + a_i + log(e_i) \quad \text{and} \quad a_i \sim N(0, \sigma^2)$$

MALMUS fits the model using `gllammm` (page 724). Using Stata's `mepoisson` we get the same results using the comand

```
mepoisson o, offset(lne) || county:
```

Note that the offset has to be specified as an option in the fixed part of the model. The model can also be fit using R as shown in the computing logs.

Using mean-variance adaptive Gauss-Hermite quadrature with 12 points we get $\hat{\alpha} = 0.0803$ and $\hat{\sigma}^2 = 0.5847$.

The average SMR in this model is 145, obtained by noting that

$$E(Y_i/e_i) = exp(\alpha + a_i) \quad \text{and} \quad E(exp(a_i)) = exp(\sigma^2/2)$$

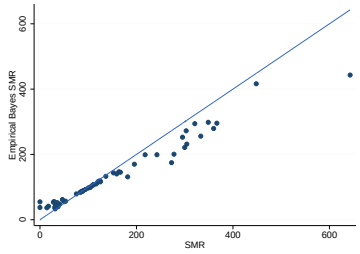
There is, however, substantial variation across counties.

Prediction of SMRs by County

We now consider predicting the SMR in each county using EB posterior means or modes. Stata's `mepoisson` uses means, but has an option for modes; `gllamm` uses means, and R uses modes.

We first predict the random effects using `predict a, reffects` or `ranef()` to obtain $\hat{a}_i$ for each county, and then add the constant but leave out the offset, computing the predicted SMR as $100 \exp\{\hat{\alpha} + \hat{a}_i\}$.

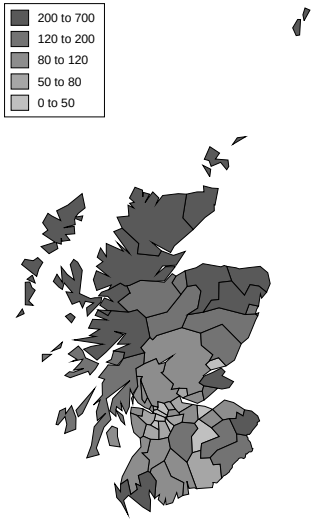
The figure on the right shows the EB estimates plotted against the crude SMRs and exhibits the usual shrinkage towards the overall mean, see MALMUS figure 13.3.



A Choropleth Map

The map on the right shows the counties of Scotland with shading representing the EB estimate of the SMR, reproducing MALMUS Figure 13.2.

The computing log shows how to reproduce this graph using Stata code available from Stata press or our own R code. The incidence of lip cancer is higher in coastal places, particularly in the north.



Health-Care Reform in Germany

MALMUS examines the extent to which the health-care reform in Germany reduced the number of doctor visits, using panel data for women working full time before and after the reform.

Here is a comparison of effect estimates from three models, all including controls for age, education, married, bad-health, log-income and summer

Model	Poisson	R-Intercept	R-Slope
Reform	0.8690	0.9547	0.9023
σ_a	-	0.9051	0.9541
σ_b	-	-	0.9303

The random-intercept model shows substantial unobserved heterogeneity in doctor visits among women with the same observed attributes; a one std dev increase in "frailty" results in 2.5 times as many visits.

The random-slope model allows the effect of the reform to vary across women. The effect for the average woman is now a 10% reduction, but varies substantially across women. The correlation between intercept and slope is -0.491.

Infant and Child Mortality in Kenya

An important application of Poisson models is to multilevel survival analysis via the connection with piecewise exponential survival.

I illustrate this approach with an analysis of infant and child mortality using the Kenya DHS, with an abridged version in "Multilevel Models in Demography" and full details in my chapter of the *Handbook of Multilevel Analysis*.

Let $\lambda_{ijk}(t)$ denote the hazard at age t for the i-th child of the j-th mother in the k-th community. We consider a three-level model

$$\lambda(t|x_{ijk}, a_{jk}, a_k) = \lambda_0(t) \exp\{x'_{ijk}\beta + a_{jk} + a_k\}$$

where $\lambda_0(t)$ is the baseline hazard, β is a vector of fixed parameters representing effects of observed covariates, and $a_{jk} \sim N(0, \sigma_2^2)$ and $a_k \sim N(0, \sigma_3^2)$ are random effects representing unobserved family and community frailty.

Estimation Using Poisson Regression

We assume the hazard is constant in intervals with cutpoints τ_d. After some exploratory work I chose cutpoints 0,1,6,12,24 and 60 months. I then split each observation into one episode per interval visited, and count events and exposure, obtaining 48,094 episodes.

Predictors include one variable at the community level (urban or rural), one at the mother level (years of education) and five at the level of the child, all well-known risk factors (gender, cohort, age of mother, birth order, and length of the previous birth interval). I'll show how these are represented when I display the coefficients.

To fit the piecewise exponential model we treat the death indicator as Poisson with the log of exposure time as an offset. Estimation using mean-variance adaptive Gaussian quadrature is implemented in Stata's `mepoisson`. (Unfortunately R's `glmer` in the `lme4` package uses PQL for three-level models. Fortunately there is a good interface to Stan for Bayesian estimation.)

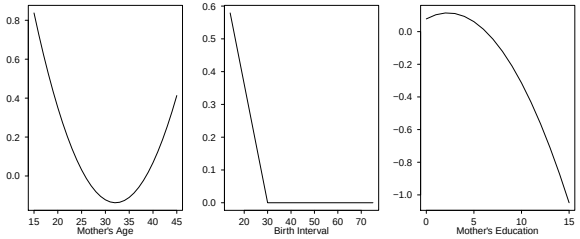
Parameter Estimates

Variable	Term	Coefficient	Standard Error	Hazard Ratio
		Fixed Coefficients		
Constant	1	-4.588	0.118	-
Age	1–5	-1.642	0.089	0.194
	6–11	-1.998	0.097	0.136
	12–23	-2.822	0.106	0.059
	24–59	-3.362	0.109	0.026
Sex	male	0.087	0.068	1.091
Cohort	1993+	0.173	0.069	1.189
Mother's	$a - 25$	-0.047	0.011	0.954
age	$(a - 25)^2$	0.003	0.001	1.003
Birth	$o - 3$	0.043	0.039	1.044
order	$(o - 3)^2$	0.004	0.005	1.004
Interval	$(30 - i)_+$	0.036	0.006	1.037
Mother's	$e - 7$	-0.068	0.015	0.934
education	$(e - 7)^2$	-0.007	0.003	0.993
Residence	urban	0.040	0.142	1.041
		Variance Parameters		
Family	σ_2	0.613	0.086	1.846
Community	σ_3	0.680	0.055	1.973

Hazard Ratios

The fixed coefficients can be interpreted in the usual fashion. Children born after 1993 have 19% *higher* risk that those born earlier, after adjusting for all other factors.

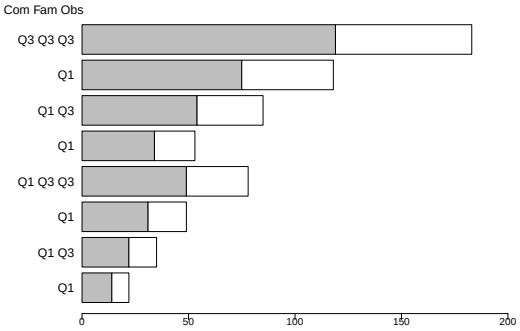
For variables represented using a quadratic or a spline a graph is always helpful:



The most remarkable feature of the results, however, is the extent to which we have unobserved heterogeneity at the family and community level.

Predicted Probabilities

A nice way to present results is to compute conditional and marginal probabilities of death by age one and five.
Here are estimated conditional (or subject-specific) probabilities for quartiles 1 and 3 of observed and unobserved risk:



The marginal (or population-average) probabilities can be obtained using Gauss-Hermite quadrature.

Multilevel Models

10. Models for Overdispersed Count Data

Germán Rodríguez

Princeton University

April 25, 2018

Negative Binomial

Count data often exhibit *overdispersion* relative to a Poisson model, in the sense that the variance exceeds the mean.

A solution is to add a multiplicative gamma random effect at level one, with mean one and variance σ^2. This results in a negative binomial model, for which the mean and variance are

$$E(Y) = \mu \quad \text{and} \quad \text{var}(Y) = \mu(1 + \sigma^2\mu)$$

The variance here is a quadratic function of the mean.

The model can be extended to multiple levels by adding additional normal random effects in the log scale.

I often find, however, that this is overkill, as multilevel Poisson models already allow overdispersion.

Software Notes: Negative Binomial

Stata can fit random-intercept negative binomial models using `xtnbreg` and more general random-coefficient negative binomial models using `menbreg`.

In R there is a `glmer.nb()` function that extends `glmer()` to negative binomial models, using adaptive quadrature for random-intercept models and PQL for models with more than one random effect.

In addition, `rstanarm` has a `stan_glmer.nb()` function to fit these models using Hamiltonian Monte Carlo (HMC).

In the health-reform data a random-intercept NB model gives results similar to the Poisson model, and a random-slope model where the reform coefficient varies randomly turns out not to be identified, resulting in a reform variance of zero (even if you restrict the fit to women observed both times).

Excess Zeroes

Another frequent occurrence with count data is to observe an excess of zeroes compared to the Poisson standard. For example in the health reform data 30% of the observations have no doctor visits, whereas a simple Poisson model predicts only 11%.

A negative binomial model often helps improve matters. In the health reform data, using a negative binomial model predicts 31% with no visits, a much better fit. The random-slope model considered in the previous unit also predicts about 30% zero visits.

There are, however, two specialized models that introduce an additional equation to take care of the excess zeroes: zero-inflated and hurdle models.

Zero-Inflated Poisson

The zero-inflated Poisson model introduced by Lambert (1992) postulates the existence of a latent class where the outcome is always zero, and another class where the outcome is drawn from a Poisson distribution.

The model uses a logit equation to predict membership in the "always zero" class, and a log-linear equation for the mean of the Poisson distribution. Both can include covariates, and the model produces structural and random zeroes.

There is also a zero-inflated negative binomial model, but again I find that this is overkill, as either zero-inflation or the level-one random effect can often model the excess zeroes.

The model can be extended in principle to a multilevel setting, adding random intercepts and slopes.

Software Notes: Zero-Inflated

Single-level zero-inflated models can be fit in Stata using `zip` for Poisson and `zinb` for negative binomial.

In R I recommend the `pscl` package, which has a `zeroinf()` function, with a `dist` argument to specify the distribution as the default "poisson" or "negbin".

There are no packaged procedures in Stata or R for zero-inflated multilevel models, but these may be programmed in Stan.

Hurdle Models

An alternative approach uses two separate models:

- a logit model to distinguish zero and positive counts, and
- a zero-truncated Poisson model to represent the counts conditional on them exceeding zero.

One can also use a negative binomial distribution for the second step, but again I find that this is often overkill.

In this model there is only one kind of zero, which makes the distinction between zero and one or more clearer. Unfortunately the coefficients no longer have a simple interpretation in terms of relative effects on the mean, because the mean of the truncated part is $\mu/(1-e^{-\mu})$ rather than μ. But one can always compute marginal effects.

Hurdle models can be extended to a multilevel setting by adding Gaussian random intercepts or slopes.

Software Notes: Hurdle Models

Fitting single-level hurdle models is easy because you fit separate logit and zero-truncated Poisson or negative binomial models.

In Stata the commands are `logit` and `tpoisson` (which supersedes `ztp`) for Poisson or `tnbreg` (which supersedes `ztnb`) for negative binomial.

In R you may use `glm()` for the Bernoulli part and the `VGAM` package, which has a function `vglm()` with a `family` argument that can be "pospoisson" or "posnegbinomial" for the truncated count portion.

Once again there are no packaged procedures in Stata or R for multilevel versions of hurdle models (or even the truncated count equation), but they can be programmed in Stan.

A Random-Intercept Hurdle Model

Here is the model we developed in class to fit a random-intercept hurdle model to the health reform data. We start with the data and parameters blocks:

```
dr_code = '
data {
  int N ;              // nobs
  int y[N];            // outcome
  int K;               // number of predictors
  row_vector[K] x[N];  // predictors
  int M;               // number of groups
  int g[N];            // mapping
  vector[2] Zero;      // mean of ri
}
parameters {
  real alpha1;          // logit equation
  vector[K] beta1;
  real alpha2;          // truncated-poisson equation
  vector[K] beta2;
  vector[2] u[M];            // random intercepts
  vector<lower=0>[2] sigma;  // st deviations of ri
  corr_matrix[2] Omega;      // correlation of ri
}
```

The model continues in the next slide.

The Model

Next we write a block to compute the covariance of the random effects and define the model, including the priors and likelihood

```
transformed parameters {
  cov_matrix[2] V;
  V = quad_form_diag(Omega, sigma);
}
model {
  alpha1 ~ normal(0,10);
  beta1 ~ normal(0,10);
  alpha2 ~ normal(0,10);
  beta2 ~ normal(0,10);
  u ~ multi_normal(Zero, Omega);
  for(n in 1:N) {
    (y[n] == 0) ~ bernoulli_logit(alpha1 + u[g[n]][1] + x[n] * beta1);
    if(y[n] > 0)
      y[n] ~ poisson(exp(alpha2 + u[g[n]][2] + x[n] * beta2))T[1,];
  }
}
'
```

The Bernoulli term contributes to the likelihood p for zeros and $1 - p$ for positive counts, and the Poisson term contributes a zero-truncated Poisson density for positive counts.

Fitting The Model

We read the data from the website, create a list and run the model

```
library(foreign)
dr <- read.dta("http://data.princeton.edu/pop510/drvisits.dta")
map <- function(id) { f <- table(id); rep(1:nrow(f), f) }
xvars = c("reform","age","educ","married","badh","loginc","summer")
dr_data <- list(N=nrow(dr), K=length(xvars), y = dr$numvisit,
  x = dr[,xvars], M = length(unique(dr$id)), g = map(dr$id),
  Zero = c(0,0))
library(rstan)
hri <- stan(model_code=dr_code, data=dr_data, chains=1, iter=1000)
```

The test run takes about one hour. The fixed effects look alright:

```
print(hri, pars=c("beta1[1]", "beta2[1]","sigma","Omega[1,2]"),probs=c(.025,.975),digits_summary=3)
...
             mean se_mean    sd   2.5%  97.5% n_eff  Rhat
beta1[1]    0.221   0.004 0.116  0.004  0.439  1000 1.010
beta2[1]   -0.015   0.001 0.036 -0.086  0.054  1000 0.999
sigma[1]    1.295   0.063 0.170  0.997  1.672     7 1.306
sigma[2]    0.787   0.006 0.032  0.728  0.856    28 1.076
Omega[1,2] -0.555   0.074 0.181 -0.856 -0.199     6 1.377
...
```

Unfortunately the results for the random effects are terrible, indicating lack of convergence and an effective sample size for the correlation of just 6!

An Alternative Model

I conclude that it is hard to estimate separate propensities for zero and positive counts. A simpler model postulates a single standard normal propensity z to visit a doctor. The logit equation has a term $\sigma_1 z$ to affect the probability of one or more visits, and the Poisson equation has a term $\sigma_2 z$ to affect the parameter μ.

This model runs in just about half an hour and yields sensible results:

```
> print(hgr, pars=c("beta1[1]", "beta2[1]", "sigma"),probs=c(.025,.975),digits_summary=3)
...
           mean se_mean    sd   2.5% 97.5% n_eff  Rhat
beta1[1] -0.189   0.003 0.102 -0.389 0.012  1000 0.999
beta2[1] -0.018   0.001 0.037 -0.087 0.056  1000 0.999
sigma[1]  0.917   0.010 0.142  0.647 1.198   192 1.013
sigma[2]  0.811   0.002 0.032  0.748 0.874   174 1.007
...
```

The reform has a large effect on whether women visit a doctor, and no effect on the number of visits of those who do. It would probably be worth running two longer chains to confirm the results.

Multilevel Models

11. Models for Ordinal Data

Germán Rodríguez

Princeton University

April 30, 2018

Categorical Data

Our final week deals with multilevel models for categorical data. We will consider ordered logit models first, which are simpler, and then turn our attention to multinomial logit models.

MALMUS notes that at the time of writing there were no official Stata commands for fitting multilevel models to categorical data other than binary, but version 14 solved the problem for ordered logits with `meologit`. As for multinomial logit models, it turns out that they can be fit as structural equation models with `gsem`, as noted by a Stata blogger.

On the R ecology I haven't found any package to fit multilevel ordered or multinomial logit models by maximum likelihood, but there are plenty of Bayesian solutions. We will use this opportunity to gather a bit more experience using Stan.

Ordered Logit Models

Recall than in an ordered logit model we focus on the logit of *cumulative* probabilities, so given an outcome Y_{ij} for the j-th observation in group i a random-intercept model would be

$$\Pr\{Y_{ij}|a_i > k\} = \text{logit}^{-1}(a_i + \boldsymbol{x}'_{ij}\boldsymbol{\beta} - \theta_k)$$

where $a_i \sim N(0, \sigma_a^2)$ is a normally-distributed random effect with mean 0 and variance σ_a^2.

The model may also be written in terms of a latent variable following a linear model

$$Y^*_{ij} = a_i + \boldsymbol{x}'_{ij}\boldsymbol{\beta} + e_{ij}$$

where e_{ij} is standard logistic and $Y_{ij} > k \iff Y^*_{ij} > \theta_k$, so the θ's may be interpreted as threshold parameters.

The equivalence follows from substituting the latent variable in $\Pr\{Y^*_{ij} > \theta_k\}$ and using the symmetry of the logistic distribution.

Treating Schizophrenia

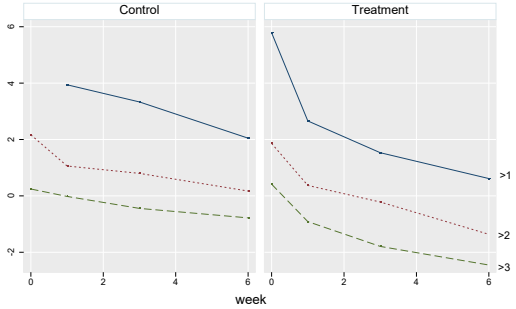
We'll analyze the example in MALMUS, a randomized trial comparing four drugs and a placebo and measuring the severity of illness using the Inpatient Multidimensional Psychiatric Scale (IMPS) at various intervals since randomization.

We combine all four drugs in a single "treated" group and recode the outcome into four severity categories: normal or borderline (≤ 2.4), moderately ill ($2.5 - 4.4$), markedly ill ($4.5 - 5.4$) and severely ill ($5.5 - 7$), as done in the original analysis.

As always, it pays to examine the data before analysis. Patients can be seen for up to seven weeks, but the most common pattern has observations in weeks 0, 1, 3 and 6. In fact no patient has more than 4 assessments.

Plotting Cumulative Proportions

A useful diagnostic plot shows the empirical logits of the proportions above each response category by week. Because weeks 2, 4 and 5 have very few assessments we omit them from the plot.



The graph shows that the treatment is generally beneficial but the trajectories are not linear. We will follow the original authors and work with the square root of weeks as the time scale.

Ordered Logits

Obviously we will need to interact treatment and time to capture treatment effects on the trajectory of each patient.

Here is a baseline ordered logit model representing population average effects (with uncorrected standard errors)

```
Ordered logistic regression                     Number of obs     =      1,603
                                                LR chi2(3)        =     501.26
                                                Prob > chi2       =     0.0000
Log likelihood = -1878.0969                     Pseudo R2         =     0.1177

------------------------------------------------------------------------------
       impso |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    sqrtweek |  -.5366467    .110815    -4.84   0.000    -.7538401   -.3194534
   treatment |  -.0006043   .1883287    -0.00   0.997    -.3697218    .3685132
 interaction |  -.7509692   .1276787    -5.88   0.000    -1.001215   -.5007235
-------------+----------------------------------------------------------------
       /cut1 |  -3.807279   .1898591                     -4.179396   -3.435162
       /cut2 |  -1.760167   .1702695                     -2.093889   -1.426445
       /cut3 |  -.4221112   .1636329                     -.7428258   -.1013965
------------------------------------------------------------------------------
```

Keep these in mind for comparison, as we move to models with subject-specific effects.

Random-Intercept Ordered Logits

Next we add a patient-specific random intercept, assumed independent of the covariates across patients.

```
meologit impso weeksqrt treatment interact || id:

Mixed-effects ologit regression                 Number of obs     =      1,603
Group variable:                id               Number of groups  =        437
  ...
Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(3)      =     480.06
Log likelihood = -1701.3811                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
       impso |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    weeksqrt |  -.7657629   .1307697    -5.86   0.000    -1.022067    -.509459
   treatment |  -.0603847   .3136873    -0.19   0.847    -.6752006    .5544311
    interact |  -1.206126   .1526656    -7.90   0.000    -1.505345   -.9069068
-------------+----------------------------------------------------------------
       /cut1 |  -5.860997   .3321236   -17.65   0.000    -6.511947   -5.210046
       /cut2 |  -2.828207   .2901595    -9.75   0.000     -3.39691   -2.259505
       /cut3 |  -.7103887   .2749679    -2.58   0.010    -1.249316   -.1714614
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   3.773713   .4650158                      2.964009    4.80461
------------------------------------------------------------------------------
LR test vs. ologit model: chibar2(01) = 353.43        Prob >= chibar2 = 0.0000
```

This model yields an intra-class correlation of 0.53 in the latent scale.

Interpreting Random Intercept Results

The treatment coefficient reflects initial differences and it is reassuringly small and not significant.

The interesting coefficient is the interaction, which exponentiated is 0.299. This indicates that the odds of begin above category 1, 2 or 3 of the IMPS are 70% lower in the treatment than in the control group at any week after randomization.

The standard deviation of the random effect indicates very substantial variation across patients, with the odds of being above any category increasing seven-fold as we move up one standard deviation from the mean with everything else the same.

We can also compute a median odds ratio $\exp\{\sqrt{2}\sigma_a\Phi^{-1}(3/4)\}$ as 6.37. This means that if we draw at random two patients with the same covariates, the ratio of the odds of scoring above any given category, when we compare the larger to the smaller odds, would exceed 6.37 half the time.

Random-Slope Ordered Logits

The next model allows the slope of the time variable to vary randomly across patients. As usual we specify an unstructured covariance matrix.

```
meologit impso weeksqrt treatment interact || id: weeksqrt, covariance(unstructured)
...
                                                Wald chi2(3)      =     254.29
Log likelihood =   -1662.73                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
             impso |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------+----------------------------------------------------------------
          weeksqrt |  -.8821765   .2175176    -4.06   0.000    -1.308503   -.4558499
         treatment |   .0525632   .3898986     0.13   0.893    -.7116241    .8167505
          interact |  -1.695097   .2520524    -6.73   0.000    -2.189111   -1.201084
-------------------+----------------------------------------------------------------
             /cut1 |   -7.32517   .4727348   -15.50   0.000    -8.251714   -6.398627
             /cut2 |  -3.423091   .3857357    -8.87   0.000    -4.179119   -2.667062
             /cut3 |  -.8174723   .3506013    -2.33   0.020    -1.504638   -.1303064
-------------------+----------------------------------------------------------------
id                 |
      var(weeksqrt)|   2.009688   .4179082                      1.336977    3.020879
         var(_cons)|   6.993466   1.313759                      4.839381    10.10637
-------------------+----------------------------------------------------------------
id                 |
cov(_cons,weeksqrt)|  -1.504658   .5300824    -2.84   0.005      -2.5436   -.4657153
------------------------------------------------------------------------------
LR test vs. ologit model: chi2(3) = 430.73                Prob > chi2 = 0.0000
```


Interpreting Random Slope Results

A comparison with the previous model yields a chi-squared of 77.24. Although the test is conservative (because we are on a boundary of the parameter space) it is clearly highly significant.

The patient-specific odds ratio per unit of time is estimated as 0.41 in the control group and 0.07 in the treated group. Both the intercept and slope vary substantially across patients with a correlation of -0.40.

As MALMUS notes, this means that patients having more severe schizophrenia at the start of the study tend to have a greater decline in severity than those with less severe schizophrenia in both the control and treatment groups.

We'll leave as an exercise computing subject-specific and population-average predicted probabilities by treatment and week.

Fitting the Models in R

We now fit exactly the same models in R. I will not repeat the graphs, but note that we can fit the standard proportional odds logistic regression model using the function `polr` in the `MASS` package. Given a data frame called `sch` the call is:

```
podds <- polr(impso ~ weeksqrt * treatment, data = sch)
 ...
> summary(podds)
...
Coefficients:
                       Value Std. Error   t value
weeksqrt           -0.5366419     0.1108 -4.842684
treatment          -0.0005995     0.1883 -0.003183
weeksqrt:treatment -0.7509752     0.1277 -5.881755

Intercepts:
                    Value    Std. Error t value
(0,2.4]|(2.4,4.4]    -3.8073   0.1899   -20.0532
(2.4,4.4]|(4.4,5.4]  -1.7602   0.1703   -10.3375
(4.4,5.4]|(5.4,7]    -0.4221   0.1636    -2.5796

Residual Deviance: 3756.194
AIC: 3768.194
```

It is reassuring to see that we have the same results as in Stata. We now try Stan.

Ordered Logit Model in Stan

We'll build the model in steps, starting from the standard ordered logit model.

```
sch_code = '
data {
  int N;  // number of observations
  int K;  // number of response categories
  int D;  // number of predictors
  int<lower=1, upper=K> y[N]; // outcomes
  row_vector[D] x[N];    // predictors
}
parameters {
  ordered[K-1] theta;
  vector[D] beta;
}
model {
  for(n in 1:N) {
    y[n] ~ ordered_logistic(x[n] * beta, theta);
  }
}'
```

The code follows the Stan manual and is remarkably simple thanks to the fact that there is an `ordered` data type to handle the thresholds and an `ordered_logistic` distribution to take care of converting the tail probabilities into a multinomial distribution.

Bayesian Ordered Logit Estimates

The next step was to put the data in a list and run Stan

```
sch_data <- list(N = nrow(sch), K = 4, D = 3,
  y = as.numeric(sch$impso), x = as.matrix(sch[,c("treatment","weeksqrt","interaction")]))
ologit <- stan(model_code=sch_code, model_name="ologit", data=sch_data, iter=2000, chains=2)
```

I specified a few options to print the results in a convenient way

```
> print(ologit, digits_summary=3, probs=c(0.025,0.5,0.975))
Inference for Stan model: ologit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

              mean se_mean    sd      2.5%       50%     97.5% n_eff  Rhat
theta[1]    -3.820   0.008 0.191    -4.196    -3.825    -3.445   572 1.000
theta[2]    -1.766   0.007 0.173    -2.084    -1.769    -1.419   554 0.999
theta[3]    -0.423   0.007 0.167    -0.742    -0.424    -0.080   528 1.000
beta[1]      0.004   0.008 0.193    -0.364    -0.003     0.389   518 1.000
beta[2]     -0.537   0.005 0.111    -0.739    -0.538    -0.315   554 0.999
beta[3]     -0.757   0.005 0.129    -1.011    -0.757    -0.507   555 1.000
lp__     -1880.036   0.065 1.687 -1884.097 -1879.734 -1877.659   671 1.004

Samples were drawn using NUTS(diag_e) at Sat Apr 23 14:47:54 2016.
```

The Bayesian estimates are very similar to the maximum likelihood estimates obtained earlier, so we soldier on.

Specifying a Random Intercept Model

Things get more interesting when we add a random intercept at the patient level. We assume that $a_i \sim N(0, \sigma)$ with a $U(0, 100)$ prior on σ and the default priors on everything else.

```
    sch_code = '
    data {
      int N;  // number of observations
|     int M;  // number of groups
      int K;  // number of response categories
      int D;  // number of predictors

      int<lower=1, upper=K> y[N]; // outcomes
      row_vector[D] x[N];         // predictors
|     int g[N];                   // map observations to groups
    }
    parameters {
      ordered[K-1] theta;
      vector[D] beta;
|     real a[M];
|     real<lower=0, upper=10> sigma;
    }
    model {
|     a ~ normal(0, sigma);
      for(n in 1:N) {
|       y[n] ~ ordered_logistic(x[n] * beta + a[g[n]], theta);
      }
    }'
```

A bar on the left margin marks new or changed lines.

Additions for Random Intercept Model

The changes to the code include

- adding the number of groups and a map to the data block
- adding the group random effects and σ_a to the parameters
- defining the prior for the random effects and modifying the linear predictor

The code assumes that the group id's are consecutive integers, which is not the case in this dataset. I wrote the following general function to map group id's when they are not the integers `1:M`:

```
map_groups <- function(id) {
  f <- table(id)
  rep(1:nrow(f), f)
}
```

And we can then add the map to the list

```
sch_data$g = map_groups(sch$id))
```


Running the Random Intercept Model

We can now run the model and (eventually) print the results. I specify the parameters to be printed to omit the random effects

```
riologit <- stan(model_code=sch_code, model_name="riologit", data=sch_data, iter=2000, chains=2)
...
print(riologit, digits_summary=3, probs=c(0.025,0.5,0.975),
      pars=c("theta[1]","theta[2]","theta[3]","beta[1]","beta[2]","beta[3]","sigma"))
Inference for Stan model: riologit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

           mean se_mean    sd   2.5%    50%  97.5% n_eff  Rhat
theta[1] -5.882   0.018 0.329 -6.553 -5.873 -5.273   351 1.007
theta[2] -2.834   0.015 0.290 -3.420 -2.822 -2.288   383 1.004
theta[3] -0.703   0.013 0.273 -1.251 -0.694 -0.199   433 1.003
beta[1]  -0.771   0.005 0.130 -1.032 -0.772 -0.519   629 1.000
beta[2]  -0.043   0.015 0.308 -0.651 -0.035  0.530   409 1.003
beta[3]  -1.210   0.006 0.150 -1.503 -1.208 -0.915   549 1.002
sigma     1.965   0.007 0.119  1.741  1.963  2.205   287 1.014

Samples were drawn using NUTS(diag_e) at Sat Apr 23 15:13:58 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

One again the results are very similar to the maximum likelihood estimates, so we are encouraged to continue.

Specifying a Random Slope Ordered Logit Model

The final step is to add a random slope. Here's the new code:

```
sch_code = '
data {
  int N;  // number of observations
  int M;  // number of groups
  int K;  // number of response categories
  int D;  // number of predictors

  int<lower=1, upper=K> y[N]; // outcomes
  row_vector[D] x[N];         // predictors
  int g[N];                   // map observations to groups
  vector[2] Zero;             // means of random effects
}
parameters {
  ordered[K-1] theta;
  vector[D] beta;
  vector[2] u[M];
  corr_matrix[2] Omega;
  vector<lower=0>[2] sigma;
}
transformed parameters {
  cov_matrix[2] Sigma;
  Sigma <- quad_form_diag(Omega, sigma);
}
model {
  u ~ multi_normal(Zero, Sigma);
  for(n in 1:N) {
    y[n] ~ ordered_logistic(x[n] * beta +
      u[g[n]][1] + u[g[n]][2]*x[n][1], theta);
  }
}'
```


Additions for Random Slope Ordered Logit Model

The basic idea is that we now have bivariate normal random effects

$$u = \begin{pmatrix} a \\ b \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_a^2, \sigma_{ab} \\ \sigma_{ab}, \sigma_b^2 \end{pmatrix} \right)$$

with an unstructured covariance matrix. One way to parametrize the variance-covariance matrix is in terms of non-negative standard deviations σ_a, σ_b and a correlation matrix, which is what we do with `sigma` and `Omega`.

We then define a *transformed parameter* to obtain the 2x2 covariance matrix `Sigma`, which can be computed from the standard deviations and correlations using the function `quad_form_diag()`.

All that remains then is to sample the bivariate random effects from a multivariate normal distribution and add them to the linear predictor, remembering to multiply the slope by the time variable.

Running the Random Slope Ordered Logit Model

We add a vector of zeroes to the data and run the model

```
sch_data$Zero <- c(0,0)
rsologit <- stan(model_code=sch_code, model_name="rsologit", data=sch_data, iter=2000, chains=2)
```

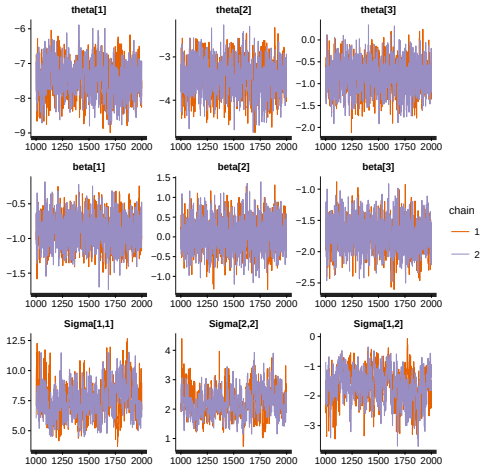
When it's all done we print the results

```
Inference for Stan model: rsologit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

             mean se_mean    sd   2.5%    50%  97.5% n_eff  Rhat
theta[1]   -7.454   0.034 0.489 -8.415 -7.443 -6.542   211 1.008
theta[2]   -3.485   0.018 0.393 -4.255 -3.477 -2.726   457 1.004
theta[3]   -0.839   0.013 0.359 -1.553 -0.825 -0.120   792 1.001
beta[1]    -0.892   0.008 0.221 -1.308 -0.892 -0.465   808 1.002
beta[2]     0.055   0.013 0.399 -0.713  0.060  0.882   965 1.000
beta[3]    -1.735   0.008 0.253 -2.227 -1.732 -1.234   941 1.000
Sigma[1,1]  7.482   0.129 1.381  5.067  7.411 10.453   114 1.016
Sigma[1,2] -1.647   0.055 0.558 -2.872 -1.616 -0.671   102 1.018
Sigma[2,1] -1.647   0.055 0.558 -2.872 -1.616 -0.671   102 1.018
Sigma[2,2]  2.198   0.049 0.470  1.364  2.169  3.213    93 1.012

Samples were drawn using NUTS(diag_e) at Sat Apr 23 17:03:40 2016.
```

One more time the results are similar to the maximum likelihood estimates.

Trace Plots for Random Slope Model

```
traceplot(rsologit, pars=c("theta[1]","theta[2]","theta[3]","beta[1]","beta[2]","beta[3]",
    "Sigma[1,1]","Sigma[2,2]","Sigma[1,2]"))
```



Multilevel Models

12. Models for Nominal Data

Germán Rodríguez

Princeton University

May 2, 2018

Multinomial Logit Model

Recall the multinomial logit model, where the probability of falling in category k for individual i is

$$\Pr\{Y_i = k\} = \frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}}{\sum_v e^{\boldsymbol{x}_i'\boldsymbol{\beta}_v}}$$

To identify the model we choose a category as reference and set $\beta_r = 0$ so the model has $K - 1$ sets of coefficients and is identified.

The k-th linear predictor $\boldsymbol{x}_i'\boldsymbol{\beta}_k$ is the log-relative probability of category k relative to r (also called the log-odds of k over r).

The model can be interpreted in terms of random utilities where the utility of choice k for individual i follows the linear model

$$U_{ik} = \boldsymbol{x}_i'\boldsymbol{\beta}_k + e_{ik}$$

where the e_{ik} are i.i.d. extreme value and $U_{ir} = 0$ serves as a baseline. Maximizing the expected utility leads to choosing category k with the probability given above.

Random-Intercept Multinomial Logits

We now extend the model to two-level data so Y_{ij} is the outcome for individual j in group i. We introduce K random intercepts per group, so the conditional probability of falling in category k is

$$\Pr\{Y_i = k|\boldsymbol{a}_i\} = \frac{e^{a_{ik}+\boldsymbol{x}_i'\boldsymbol{\beta}_k}}{\sum_v e^{a_{iv}+\boldsymbol{x}_i'\boldsymbol{\beta}_v}}$$

but set $a_{ir} = 0$ so we are left with $K - 1$ random effects assumed to have a multivariate normal distribution with mean vector zero and arbitrary variance-covariance matrix.

The log-relative conditional probability of category k over r given the random effects a_{ik} for $k \neq r$ is then

$$\log \frac{\Pr\{Y_{ik}|\boldsymbol{a}_i\}}{\Pr\{Y_{ir}|\boldsymbol{a}_i\}} = a_{ik} + \boldsymbol{x}_i'\boldsymbol{\beta}_k$$

so a_{ik} can be interpreted as a latent propensity to choose category k over r net of the covariates.

The McKinney Homeless Study

We will illustrate the methods using the McKinney Homeless study, which has generated interesting longitudinal data on 361 at-risk individuals randomly assigned to one of two types of case management (comprehensive vs. traditional) and one of two levels of access to independent housing using "Section 8" certificates.

The outcome is housing status at baseline and at 6, 12 and 24 months, classified as streets/shelters, community housing, or independent housing. The predictors of interest include time and `sec8`, a dummy variable coded one for the treatment group.

The data have been analyzed by Don Hedeker, author of the `mixno` package* for fitting mixed multinomial models using Gauss-Hermite quadrature. We will fit essentially the same model, although for simplicity we will treat time (coded 0 to 3) linearly instead of using dummy variables.

*`https://www.jstatsoft.org/article/view/v004i05`

Population Average Effects

For reference purposes I fitted a standard multinomial logit model estimating population average effects (with uncorrected standard errors).

```
. mlogit status c.sec8##c.time,  base(0)
Multinomial logistic regression                 Number of obs     =      1,289
                                                LR chi2(6)        =     316.57
                                                Prob > chi2       =     0.0000
Log likelihood =   -1223.16                     Pseudo R2         =     0.1146
```

status	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
street	(base outcome)					
community						
sec8	.2395584	.2130131	1.12	0.261	-.1779395	.6570564
time	.7961881	.1075957	7.40	0.000	.5853045	1.007072
c.sec8#c.time	-.5180044	.1576099	-3.29	0.001	-.8269142	-.2090947
_cons	-.2109418	.1428502	-1.48	0.140	-.4909231	.0690395
independent						
sec8	1.157348	.2551718	4.54	0.000	.6572208	1.657476
time	1.138287	.123074	9.25	0.000	.8970665	1.379508
c.sec8#c.time	-.2308719	.1637933	-1.41	0.159	-.5519008	.090157
_cons	-1.405489	.1964876	-7.15	0.000	-1.790598	-1.02038

We'll compare these to Bayes estimates.

Maximum Likelihood via SEM

Stata does not have a command for multinomial logit models with random effects, but Rebecca Pope explains how to fit the model using structural equation models via gsem in the Stata Newsletter `http://www.stata.com/stata-news/news29-2/xtmlogit/` using

```
gsem (1.status <- sec8 time secXtime RI1[id]) ///
     (2.status <- sec8 time secXtime RI2[id]), mlogit}
```

The model defines two latent variables that vary across groups to capture random effects for each equation. The variances and covariance of these random effects are

var(RI1[id])	1.744592	.4753894			1.022697	2.976052
var(RI2[id])	3.818815	.7779019			2.56175	5.692728
cov(RI2[id],RI1[id])	1.701683	.5029326	3.38	0.001	.7159536	2.687413

This implies a correlation of 0.66 between the two latent variables representing the contrast of community over street and of independent over street.

Subject-Specific Effects

The estimates of the fixed effects are shown below

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
0.status	(base outcome)					
1.status						
sec8	.3835373	.2829853	1.36	0.175	-.1711037	.9381783
time	1.015074	.1329061	7.64	0.000	.7545826	1.275565
secXtime	-.5786798	.181297	-3.19	0.001	-.9340153	-.2233442
RI1[id]	1	(constrained)				
_cons	-.2063278	.1926945	-1.07	0.284	-.5840022	.1713466
2.status						
sec8	1.60725	.3791572	4.24	0.000	.8641157	2.350384
time	1.530787	.1579142	9.69	0.000	1.221281	1.840293
secXtime	-.2223493	.1998801	-1.11	0.266	-.6141072	.1694085
RI2[id]	1	(constrained)				
_cons	-2.047767	.3034708	-6.75	0.000	-2.642559	-1.452975

There is a trend away from the street, towards community housing in the control group and towards independent housing in the Section 8 group.

Multinomial Logit Models via Stan

Let us now explore fitting these models in a Bayesian framework using Stan. We start with the standard multinomial logit model.

There is an example in the Stan manual using one equation per outcome, a model that they note is identified only "if there are suitable priors on the coefficients". A faster and in my view preferable alternative is to work with only $K-1$ equations for K response categories, as we did for maximum likelihood.

We define the coefficients to be estimated as a $K-1$ by P matrix, and then add a row of zeroes to match the reference category in a new K by P matrix defined in the transformed parameters block.

The function used to convert multinomial logits to probabilities is called `softmax` in the machine learning literature and Stan. The newer function `categorical_logit()` calls that implicitly.

Stan Code for Multinomial Logit

```
sd_model <- '
  data {
    int K;   // number of outcome categories
    int K1;  // K-1
    int N;   // number of observations
    int P;   // number of predictors a.k.a. D
    int y[N];// outcome, coded 1 to K for each obs
    vector[P] x[N]; // predictors, including constant
  }
  transformed data {
    row_vector[P] base;
    base = rep_row_vector(0, P);
  }
  parameters {
    matrix[K1,P] beta;
  }
  transformed parameters {
    matrix[K, P] betap;
    betap = append_row(base, beta);
  }
  model {
    // prior for beta (vectorized)
    for(k in 1:K1) {
      beta[k] ~ normal(0,5);
    }
    // likelihood of outcome
    for(n in 1:N) {
      y[n] ~ categorical_logit(betap * x[n]);
    }
  }
'
```


The Stan Output

And here are the results of running the model

```
mlogit <- stan(model_code=sd_model,model="mlogit",data=sd_data,iter=2000,chains=2)

> print(mlogit, pars="beta", digits_summary=3, probs=c(0.025,0.5,0.975))
Inference for Stan model: mlogit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

           mean se_mean    sd   2.5%    50%  97.5% n_eff  Rhat
beta[1,1] -0.216   0.005 0.148 -0.503 -0.216  0.070   872 1.001
beta[1,2]  0.246   0.007 0.218 -0.184  0.249  0.662   966 1.000
beta[1,3]  0.805   0.004 0.109  0.602  0.804  1.026   933 1.004
beta[1,4] -0.525   0.005 0.159 -0.832 -0.526 -0.213   919 1.003
beta[2,1] -1.421   0.007 0.201 -1.835 -1.423 -1.044   724 1.000
beta[2,2]  1.171   0.009 0.263  0.669  1.174  1.684   878 1.000
beta[2,3]  1.151   0.005 0.124  0.906  1.154  1.386   669 1.002
beta[2,4] -0.239   0.006 0.166 -0.558 -0.241  0.086   830 1.002

Samples were drawn using NUTS(diag_e) at Tue May 01 13:47:57 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

The measures of effective sample size are all reassuring and the values of Rhat are close to 1 as they should be at convergence.

Comparison of Maximum Likelihood and Bayes

Here's a side-by-side comparison of ML and Bayes estimates by equation

Variable	Community/Street		Independent/Street	
Name	ML	Bayes	ML	Bayes
sec8	0.240	0.246	1.157	1.171
time	0.796	0.805	1.138	1.151
interaction	-0.518	-0.525	-0.231	-0.239
constant	-0.211	-0.216	-1.405	-1.421

I think the agreement is quite remarkable. We are thus encouraged to try adding random effects.

Random Intercepts in Multinomial Logits

We will add two correlated random intercepts for each individual, representing unobserved effects on the propensity to be in community and in independent housing rather than on the street. To define the model I generally followed the Stan manual.

We will define the multivariate normal distribution in terms of a vector of scale parameters and a matrix of correlations, which are the actual parameters to be estimated, just as we did for the random slope ordered logit model. This time, however, I defined the vector of means in a `transformed data` block.

In the model block we define the priors and hyper-priors for the random effects. The random effects are `multi_normal`. For the scales of the random effects I tried half `cauchy(0, 2.5)` priors, but got better results with uniforms. For the correlation I used a LKJ prior with parameter 2; for more on this prior see `http://www.psychstatistics.com/2014/12/27/d-lkj-priors/`.

Stan Code for Random-Effects Multinomial Logit

The model is long enough that I will present it in two parts. Here's Part 1 showing the data, transformed data and parameters.

```
sd_model <- '
    data {
      int K;          // number of outcome categories
      int K1;         // K-1
      int N;          // number of observations
      int P;          // number of predictors a.k.a. D
      int y[N];       // outcome, coded 1 to K for each obs
      vector[P] x[N]; // predictors, including constant
      int G;          // number of groups
      int map[N];     // map obs to groups
    }
    transformed data {
      vector[K1] zero;
      real baseline;
      zero = rep_vector(0, K1);
      baseline = 0;
    }
    parameters {
      matrix[K1,P] beta;         // fixed effects
      corr_matrix[K1] omega;     // ranef correlations
      vector<lower=0,upper=10>[K1] sigma; // ranef scales
      vector[K1] u[G];           // random intercepts
    }
...
```


Stan Code for Random-Effects Multinomial Logit

And here's Part 2, showing transformed parameters and the model block:

```
transformed parameters{
  cov_matrix[K1] V;
  V = quad_form_diag(omega, sigma);
}
model {
  // prior for beta (vectorized)
  for(k in 1:K1) {
    beta[k] ~ normal(0,5);
  }
  // prior/hyper prior for random effects
  // sigma ~ cauchy(0, 2.5);
  omega ~ lkj_corr(2);
  for(g in 1:G) {
    u[g] ~ multi_normal(zero, V);
  }
  { // local block for linear predictor
    vector[K] xb;
    for(n in 1:N) {
      xb = append_row(baseline, beta*x[n] + u[map[n]]);
      y[n] ~ categorical_logit(xb);
    }
  }
}'
```

The local block is used to add a zero to the linear predictor.

Results

This is the call used to run the model

```
rimlogit <- stan(model_code=sd_model,model="rimlogit",data=sd_data,iter=2000,chains=2)
```

And these are the results

```
> print(rimlogit, digits_summary=3, probs=c(0.025,0.5,0.975),
+       pars=c("beta","sigma","omega[1,2]"))
Inference for Stan model: rimlogit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

             mean se_mean    sd   2.5%    50%  97.5% n_eff  Rhat
beta[1,1]  -0.208   0.004 0.197 -0.602 -0.207  0.175  2000 0.999
beta[1,2]   0.363   0.006 0.291 -0.193  0.357  0.944  2000 1.001
beta[1,3]   1.018   0.005 0.133  0.770  1.016  1.281   865 1.001
beta[1,4]  -0.583   0.004 0.175 -0.919 -0.587 -0.252  2000 1.001
beta[2,1]  -2.056   0.008 0.298 -2.633 -2.048 -1.477  1260 1.000
beta[2,2]   1.584   0.009 0.371  0.872  1.576  2.301  1566 1.000
beta[2,3]   1.535   0.005 0.156  1.236  1.530  1.852  1064 1.000
beta[2,4]  -0.215   0.004 0.194 -0.616 -0.215  0.142  2000 1.000
sigma[1]    1.339   0.015 0.196  0.943  1.333  1.732   172 1.008
sigma[2]    1.965   0.011 0.198  1.620  1.957  2.378   322 1.002
omega[1,2]  0.625   0.006 0.092  0.427  0.632  0.778   254 1.000

Samples were drawn using NUTS(diag_e) at Tue May 01 10:58:04 2018.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```


Comparison of Estimates

Here are the maximum likelihood and Bayesian estimates

Variable Name	Community/Street ML	Community/Street Bayes	Independent/Street ML	Independent/Street Bayes
sec8	0.384	0.363	1.620	1.584
time	1.015	1.018	1.536	1.535
interaction	-0.579	-0.583	-0.220	-0.215
constant	-0.206	-0.208	-2.079	-2.056
scale	1.321	1.339	1.954	1.975
correlation	0.659	0.625		

The two sets of estimates are remarkably close, as one would expect from generally non-informative priors.

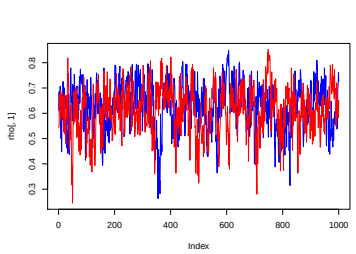
I report the posterior means of the scale parameters and correlation coefficient rather than the variances and covariance, so for comparability I translated the maximum likelihood results.

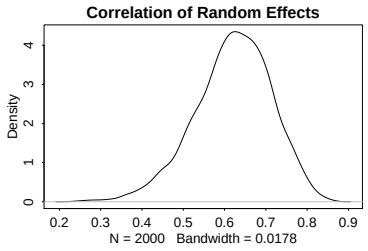
Trace Plots and Posterior Densities

We can extract any coefficient using `extract`. Here's the correlation of the random effects

```
r <- as.data.frame(extract(rimlogit, pars="omega[1,2]"), permute=FALSE)
```

And we can then do trace and/or density plots





We have a nice fuzzy caterpillar and the posterior is fairly symmetric around the mean of 0.6.

Calculating Predicted Probabilities

We can also use the samples to calculate any function of interest. Here's code to compute the predicted probabilities for the average person at time 3 under control and "Section 8" conditions.

```
> ep <- as.data.frame(extract(rimlogit, "beta"))
> names(ep)<-c("cons1","cons2","sec1","sec2","time1","time2","int1","int2")
> u0 =  cbind(0, ep[,"cons1"]+ep[,"time1"]*3, ep[,"cons2"]+ep[,"time2"]*3)
> p0 = colMeans(exp(u0)/rowSums(exp(u0)))
> u1 = u0 + cbind(0, ep[,"sec1"]+ep[,"int1"]*3, ep[,"sec2"]+ep[,"int2"]*3)
> p1 = colMeans(exp(u1)/rowSums(exp(u1)))
> rbind(p0,p1)
          [,1]      [,2]      [,3]
p0 0.03376637 0.5530774 0.4131562
p1 0.02773570 0.1153388 0.8569255
```

The probability of being in independent housing at the end of follow up for the average person is 41% in the control group and 86% in the Section 8 group, with only 3% on the street.

Try doing a trace and/or density plot, or constructing a 95% credible interval for the probability of independent housing.