



runmlwin: A Program to Run the MLwiN Multilevel Modeling Software from within Stata

George Leckie
University of Bristol

Chris Charlton
University of Bristol

Abstract

We illustrate how to fit multilevel models in the **MLwiN** package seamlessly from within **Stata** using the **Stata** program `runmlwin`. We argue that using **MLwiN** and **Stata** in combination allows researchers to capitalize on the best features of both packages. We provide examples of how to use `runmlwin` to fit continuous, binary, ordinal, nominal and mixed response multilevel models by both maximum likelihood and Markov chain Monte Carlo estimation.

Keywords: `runmlwin`, **MLwiN**, **Stata**, multilevel model, random effects model, mixed model, hierarchical linear model, clustered data, maximum likelihood estimation, Markov chain Monte Carlo estimation.

1. Introduction

Multilevel models are becoming increasingly popular in the social, behavioral and medical sciences. These models are described variously as multilevel models, random effects models, mixed models, and hierarchical linear models. A range of text books provide introductory (Hox 2010), intermediate (Raudenbush and Bryk 2002; Snijders and Bosker 2012) and advanced (Goldstein 2011) treatment of these models while two recent handbooks describe the latest developments (de Leeuw and Meijer 2008; Hox and Roberts 2010).

1.1. MLwiN software

MLwiN is a specialized software package for fitting multilevel models (Rasbash, Charlton, Browne, Healy, and Cameron 2009). At the time of going to press, the most recent version of **MLwiN** is version 2.26 and our discussion relates to this version. **MLwiN** can estimate multilevel models for continuous, binary (logistic, probit, complementary log-log), count

(Poisson, negative binomial), ordinal (ordered logistic, ordered probit) and nominal responses (unordered logistic, unordered probit). **MLwiN** provides fast estimation by both iterative generalized least squares (Goldstein 1986, IGLS), which results in maximum likelihood estimates, and by Bayesian estimation using Markov chain Monte Carlo (MCMC) methods (Browne 2012). The latter allows a range of more advanced multilevel models to be fitted, including: multilevel multivariate mixed response type models (simultaneous equation models or multiprocess models), multilevel cross-classified models (crossed random effects models), multilevel multiple membership models, multilevel spatial models (disease mapping models), multilevel measurement error models, and multilevel multiple imputation models (missing data models). The **MLwiN** Users' Manual (Rasbash, Steele, Browne, and Goldstein 2012) and **MLwiN** MCMC Manual (Browne 2012) give step-by-step instructions on how to fit these models. **MLwiN** runs under Windows and information on how to obtain the package can be found on the software's website (<http://www.bristol.ac.uk/cmm/software/mlwin/>). **MLwiN** is free to the UK academic community thanks to funding from the UK Economic and Social Research Council (ESRC). An unrestricted 30-day trial version of **MLwiN** is available to readers simply wishing to explore the software.

MLwiN is usually run interactively through a series of drop-down menus, interactive modeling windows, and dialog boxes. A unique feature of **MLwiN** is its point-and-click Equations window which allows users to specify and read the results of their multilevel models using standard mathematical notation. These features provide an excellent environment for teaching and learning multilevel modeling, but are less useful when performing multilevel analyzes for research purposes where researchers typically wish to fit a large number of models and to interpret each of these through calculating and graphing model predictions. Carrying out such an analysis using **MLwiN**'s point-and-click interface is laborious and error prone. While **MLwiN** provides a macro language for carrying out longer analyzes in a documentable and reproducible manner, the command syntax is difficult to use and is supported by limited documentation. Most **MLwiN** users stick with the point-and-click interface and will therefore not use the more advanced multilevel modeling features present in **MLwiN** as they typically require the most point-and-click steps.

1.2. Stata software

Stata (StataCorp. 2011) is a general purpose package for statistics, publication quality graphics and data management and is widely used by applied researchers, particularly those in health research and in economics. At the time of going to press, the most recent version of **Stata** is version 12.1 and our discussion relates to this version. **Stata** provides hundreds of standard and more advanced commands for statistical modeling, including commands to fit three types of multilevel model which we describe below. **Stata**'s graphics commands make it easy to generate many of the graphs popular in multilevel analysis, including: trellis plots, quantile-quantile (Q-Q) plots, 'caterpillar' plots, funnel plots, and thematic maps for spatial data. **Stata**'s data-management commands give users complete control over all types of multilevel data: you can combine and reshape large multilevel datasets, manage variables at different levels of data hierarchies, and calculate statistics across groups or replicates. **Stata** also has advanced tools for managing specialized forms of multilevel data, including panel/longitudinal data and discrete time survival data. Finally, **Stata** has an intuitive command syntax and allows one to write do-files (scripts) which makes it easy to fit long series of models and to perform associated analyzes in a documentable and reproducible manner. These features are fully documented

in the comprehensive *Stata* manual series (<http://www.stata-press.com/manuals/>), while dedicated books on statistical modeling, graphics, data management and programming in *Stata* are published in the *Stata* Press series (<http://www.stata-press.com/books/>).

Stata provides the `xtmixed`, `xtmelogit` and `xtmepoisson` commands for fitting multilevel continuous, binary (logistic only) and count (Poisson only) response models, respectively. However, if all that is required are two-level random intercept models, then the simpler and computationally faster `xtreg`, `xtlogit` and `xtpoisson` commands can be employed, respectively. An excellent introduction to all these commands is provided in [Rabe-Hesketh and Skrondal \(2012a\)](#) and [Rabe-Hesketh and Skrondal \(2012b\)](#). These commands, however, cannot fit multilevel binary probit models or multilevel negative binomial count models. There is also no provision for fitting multilevel ordinal and nominal response models. Estimation is by maximum likelihood; there is no provision for Bayesian estimation. While these three commands offer some provision for fitting cross-classified models, the implementation of these models as constrained hierarchical models is computationally inefficient and only proves feasible in models with few random effects fitted to small datasets. *Stata* also cannot fit many of the more advanced multilevel models available in **MLwiN** including multilevel multivariate mixed response type models, multilevel multiple membership models, multilevel spatial models, multilevel measurement error models and multilevel multiple imputation models.

Stata is also a programming language, and this has led many users to write their own add-on packages. In terms of multilevel analysis, the most prominent of these is the `gllamm` command ([Rabe-Hesketh, Skrondal, and Pickles 2004](#)). This command allows one to fit a much wider range of multilevel models than that provided by *Stata*'s own commands, including some models which can also not be fitted in **MLwiN**. This command can also fit many latent variable models including structural equation and latent class models. An important disadvantage of `gllamm` is that it is computationally slow, even for simple models such as those that can be fitted by *Stata*'s own commands. Using `gllamm` to fit realistically complex multilevel models with many random effects to large datasets can prove prohibitively time consuming.

There are several other user-written commands for running external software packages to perform multilevel modeling within *Stata*. These include: `sabrestata` for running the *Sabre* package for fast maximum likelihood estimation of multiprocess event/response sequences ([Crouchley, Stott, and Pritchard 2009](#)); `hlm` for running the hierarchical linear modeling (**HLM**) software ([Raudenbush, Bryk, and Congdon 2012](#)); `realcomimpute` for running the **REALCOM-IMPUTE** software for multilevel multiple imputation ([Carpenter, Goldstein, and Kenward 2011](#)); `runmplus` for running the **MPlus** software for multilevel structural equation modeling ([Muthén and Muthén 2012](#)); and `winbugs` ([Thompson, Palmer, and Moreno 2006](#)) for Bayesian model fitting in the **WinBUGS** package ([Spiegelhalter, Thomas, Best, and Lunn 2003](#)). We shall not review these commands further here.

1.3. The `runmlwin` command

We have developed the `runmlwin` command to be able to fit the full range of multilevel models available in **MLwiN** seamlessly from within *Stata*. The command allows the user to fit models by both the IGLS and MCMC algorithms and provides full control over all aspects of model specification and estimation. The `runmlwin` command works by writing, sending and then running an **MLwiN** macro file for the specified model in **MLwiN** and then returns, stores and displays the model results in *Stata* at which point the user can take full advantage of *Stata*'s

many hypothesis testing, model comparison, graphics and other post-estimation commands to aid their interpretation and analysis.

The `runmlwin` command requires Stata 9 or later and can be downloaded and installed from within Stata by issuing the following command

```
. ssc install runmlwin
```

When using `runmlwin` for the first time, users must specify the full path name of the **MLwiN** executable using a global macro called `MLwiN_path`. Users must substitute the path name that is correct for them

```
. global MLwiN_path "C:\Program Files\MLwiN v2.26\i386\mlwin.exe"
```

The rest of this article takes a practical approach to describing `runmlwin` rather than a formal description of the command's syntax. Readers seeking the latter should consult the `runmlwin` help file (Leckie and Charlton 2013).

```
. help runmlwin
```

We shall present worked examples based on two datasets analyzed in the **MLwiN** User and MCMC Manuals. Section 2 considers multilevel continuous response models while Section 3 considers multilevel binary response models. In these two sections we focus on relatively simple models which can be fitted by both `runmlwin` and Stata's own multilevel modeling commands `xtmixed` and `xtmelogit`. We do this to make our description of `runmlwin` as accessible as possible to both existing **MLwiN** and Stata users but also to readers who currently use other software. A Stata do-file to replicate all analyzes is provided in the Supplementary materials. Section 4 demonstrates a selection of more advanced models that can only be fitted by `runmlwin`. Section 5 concludes.

2. Continuous response models

In this section we consider how to fit two-level random intercept and random slope models to continuous response variables. We use the London schools' exam scores data analyzed in Chapters 2–7 of the **MLwiN** User Manual.

The data are a sub-sample of examination results from six inner London Education Authorities (school districts). The examination is the General Certificate of Secondary Education (GCSE) taken at the end of compulsory schooling, normally when students are 16 years of age. The continuous response variable is student's exam score. A key aim of the original analysis was to establish whether some schools were more effective than others in promoting students' learning and development, taking account of variations in the characteristics of students when they started secondary school (Goldstein, Rasbash, Yang, Woodhouse, Pan, Nuttall, and Thomas 1993). The data have a two-level hierarchical structure, with 4,059 students (level 1) nested within 65 schools (level 2). The data can be read into Stata by running the following command

```
. use "http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta", clear
```

The data include the following variables

- **school**: school identifier;
- **student**: student identifier;
- **normexam**: student's exam score at age 16, normalized to have a standard normal distribution;
- **cons**: equal to the number 1 for each student and is used as the intercept term in the models;
- **standlrt**: student's score at age 11 on the London reading test (LRT), standardized;
- **girl**: student's gender (0: boy; 1: girl);
- **schgend**: school's gender (1: mixed school; 2: boys' school; 3: girls' school).

2.1. Variance components model

The simplest multilevel model we might consider for these data is a two-level students-within-schools random intercept model where we make no adjustments for predictor variables. This model is referred to as a variance components model as it decomposes the response variation into separate level-specific variance components. The model is written as

$$\begin{aligned}\text{normexam}_{ij} &= \beta_0 + u_j + e_{ij} \\ u_j &\sim N(0, \sigma_u^2) \\ e_{ij} &\sim N(0, \sigma_e^2)\end{aligned}$$

where normexam_{ij} is the age 16 exam score for student i in school j , β_0 is the intercept measuring the mean score across all schools, u_j is a school level random effect, and e_{ij} is a student level residual error. The u_j and e_{ij} are assumed independent of one another and normally distributed with zero means and constant variances σ_u^2 and σ_e^2 . Multilevel models are typically described in terms of their fixed part (that part involving parameters fixed across the whole sample) and their random part (that part involving the random effects and residual error). In this very simple model, the fixed part is simply β_0 , while the random part is $u_j + e_{ij}$.

The degree of clustering (dependence) in the data can be summarized by the intraclass correlation coefficient (ICC) and the variance partition coefficient (VPC). In this simple model, the formulas for these two coefficients coincide $\sigma_u^2 / (\sigma_u^2 + \sigma_e^2)$. The ICC measures the expected correlation between two students from the same school, while the VPC measures the proportion of total variance which lies at the school level.

When specifying the `runmlwin` syntax to fit this model, it is helpful to think of each right hand side term in the above model equation as being associated with an explanatory variable which is equal to the number 1 for all observations in the data. In our data the variable `cons` plays this role and so we can rewrite the model equation as

$$\text{normexam}_{ij} = \beta_0 \text{cons} + u_j \text{cons} + e_{ij} \text{cons}$$

where `cons` appears once in the fixed part of the model, once in the random part of the model at level 2 and once in the random part of the model at level 1. The `runmlwin` command for fitting this model is then

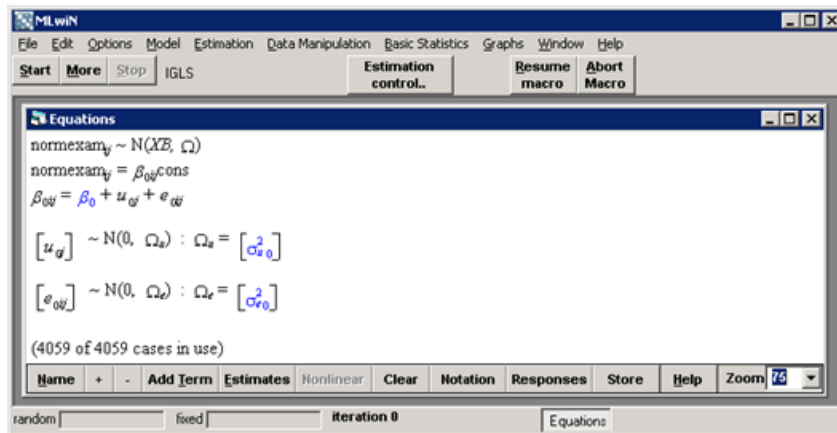


Figure 1: MLwiN software: The Equations window shows the specified model.

```
. runmlwin normexam cons, level2(school: cons) level1(student: cons)
```

While the structure of this syntax will be transparent to Stata users, it will be less clear for readers unaccustomed to Stata and so we shall discuss it in some detail. The syntax begins with the command `runmlwin` and then defines the response variable `normexam` followed by the list of predictors included in the fixed part of the model, here only `cons`. Next appears a comma which denotes the end of the fixed part of the model and the start of the random part of the model. The first term after the comma, `level2(school: cons)`, specifies the random part of the model at level 2. Within the parentheses, the level identifier, `school`, is specified before the colon while all variables made random at level 2, here only `cons`, are specified after the colon. The second term after the comma, `level1(student: cons)`, specifies the random part of the model at level 1. Again the level identifier, here `student`, is specified before the colon while all variables made random at this level, here only `cons`, are specified after the colon.

Running the above command fits the specified model by writing, sending and then running an MLwiN macro file in MLwiN. The MLwiN software automatically opens and displays the specified model in the Equations window using standard mathematical notation (Figure 1).

At this point the macro pauses and the user is presented with a choice to either click the **Resume macro** button to fit the model or to click the **Abort macro** button to terminate the macro. Users should click the **Abort macro** button if they wish to use MLwiN interactively prior to manually fitting their model or if the model displayed in the Equations window does not match the desired model. At this point we again emphasize the pedagogic value of the MLwiN Equations window as an aid to better understanding and interpreting the statistical models being specified and fitted.

In our case, the model displayed in the Equations window matches the model we wish to estimate and so we click the **Resume** button to fit the model. MLwiN then iterates until convergence. When the model has converged, the Equations window updates and displays the parameter estimates, standard errors and deviance statistic for the converged model (Figure 2).

At this point the macro pauses for a second time and the user is again presented with a choice to either click the **Resume macro** button, this time to send the model results back to Stata, or

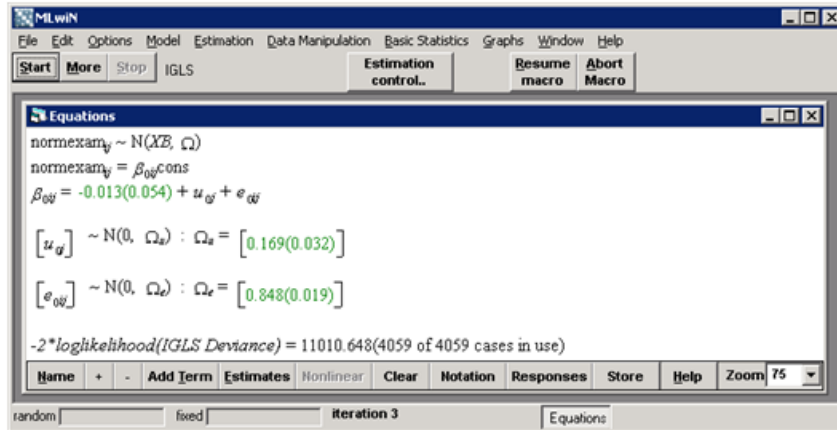


Figure 2: **MLwiN** software: The Equations window shows the parameter estimates, standard errors and deviance statistics for the converged model.

to click the **Abort macro** button to terminate the macro. Users should click the **Abort macro** button if they wish to use **MLwiN** interactively, for example to take advantage of **MLwiN**'s own post-estimation facilities. In our case, we wish to return the model results to **Stata** and so we click the **Resume macro** button. **Stata** will then return, store and display the `runmlwin` command and its associated model output in the **Stata Results** window.

```
. runmlwin normexam cons, level2(school: cons) level1(student: cons)
```

```
MLwiN 2.26 multilevel model           Number of obs       =       4059
Normal response model
Estimation algorithm: IGLS
```

Level Variable	No. of Groups	Observations per Group		
		Minimum	Average	Maximum
school	65	2	62.4	198

```
Run time (seconds) =      1.28
Number of iterations =      3
Log likelihood      = -5505.3242
Deviance            = 11010.648
```

normexam	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
cons	-.0131668	.0536254	-0.25	0.806	-.1182706	.091937

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
---------------------------	----------	-----------	----------------------	--

```

Level 2: school      |
                   | var(cons) | .1686251 .0324466 .1050309 .2322194
-----+-----
Level 1: student    |
                   | var(cons) | .8477613 .0189712 .8105786 .8849441
-----

```

The formatting of the model output follows that of other **Stata** estimation commands. The top left and right blocks of output report that **MLwiN** 2.26 was used to fit a normal response model to an estimation sample of 4,059 observations (students) using the IGLS algorithm.

The first table of output summarizes the multilevel structure of the data and informs us that the model is fitted to 65 schools where the number of students per school ranges from 2 to 198 with an average of 62.4 students. (This table plays the role of the **MLwiN Hierarchy Viewer** window.)

The next block of output reports that **runmlwin** took 1.28 seconds and 3 iterations of the IGLS algorithm to fit the model. The log-likelihood and deviance statistics are also presented.

The second table of output reports the fixed part parameters. The intercept β_0 is estimated to be -0.013 with standard error 0.054. Recall that the response is approximately normalized and so an estimated intercept of effectively zero is expected. The z ratio, p value and 95% confidence intervals are also reported, but in this example are of little substantive interest.

The final table of output reports the random part parameters. The between-school variance σ_u^2 is estimated to be 0.169 while the within-school variance σ_e^2 is estimated to be 0.848. We can calculate the ICC using the **display** command, where we refer to the parameter estimates by their internal **runmlwin** names: $\hat{\sigma}_u^2$ is referred to as **[RP2] var(cons)** while $\hat{\sigma}_e^2$ is referred to as **[RP1] var(cons)**. (The **display** command corresponds to the **CALC** macro in **MLwiN**.)

```
. display [RP2]var(cons)/([RP2]var(cons) + [RP1]var(cons))
```

```
.16590652
```

Interpreted as an ICC, the correlation in exam scores between schoolmates is 0.166. Interpreted as a VPC, 16.6% of the variation in exam scores lies between schools. There are clearly substantial differences in students' scores between schools.

Following **Stata** convention, standard errors and 95% confidence intervals are presented for the two variance components, however we caution against using them to test whether the between school differences are statistically significant. This is because z -tests and 95% confidence intervals assume asymptotic normal sampling distributions while variances are known to have positively skewed sampling distributions. The preferred way to test the between-school variance is to use **Stata's** **lrtest** command to perform a likelihood ratio test to compare the above model to a single-level model with no school effects (not shown). (The **lrtest** command corresponds to the **MLwiN Tail Areas** window.) Doing this confirms that there are indeed significant differences between schools ($\chi_1^2 = 498.72$, $p < 0.001$); students from the same school are therefore significantly more alike than students from different schools. A multilevel approach to analyze the data is clearly favored over a single-level approach.

When we ran the above **runmlwin** command, we clicked the **Resume macro** button twice in order to return the model results to **Stata**. We can add the **nopause** option to prevent

`runmlwin` from pausing in this way. When we do this the results are automatically returned to Stata. This option proves essential when performing simulation studies using `runmlwin`.

```
. runmlwin normexam cons, level2(school: cons) level1(student: cons) nopause
```

2.2. Random intercept model with predictor variables

In our next model, we adjust for students' age 11 scores, student gender and school gender. We adjust for school gender by including a boys' school dummy variable and a girls' school dummy variable (the reference category is mixed sex schools). We create these variables in Stata with the `generate` command. (The `generate` command corresponds to the `CALC` macro in `MLwiN`.)

```
. generate boysch = (schgend==2)
. generate girlsch = (schgend==3)
```

The model is written as

$$\text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + \beta_3 \text{boysch}_j + \beta_4 \text{girlsch}_j + u_j + e_{ij}$$

$$u_j \sim N(0, \sigma_u^2)$$

$$e_{ij} \sim N(0, \sigma_e^2)$$

To fit this model using `runmlwin` we simply add the four new predictors to the list of variables that appear in the `runmlwin` syntax for fixed part of the model.

When we fit multilevel models, it is often of interest to examine empirical Bayes estimates of the random effects to assess whether the random effects are (approximately) normally distributed and to make inferences about specific groups. We can retrieve the empirical Bayes estimates of the school random effects by specifying the level 2 random part of the model as `level2(school: cons, residuals(u))` where the term `residuals(u)` is an option which only applies to this part of the model. The option requests that the random effects' estimates and standard errors are returned to Stata as two new variables `u0` and `u0se` where the user can change the variable stub `u` by specifying a different string within `residuals()`. All the functionality of the `MLwiN` Residuals window (`Setting` tab) is contained within this `residuals()` option.

We spread the `runmlwin` command for this model over three lines using three slashes `///` to denote line continuation. We specify the `nogroup` option to suppress the table summarizing the multilevel structure of the estimation sample as this is the same as before.

```
. runmlwin normexam cons standlrt girl boysch girlsch, ///
> level2(school: cons, residuals(u)) ///
> level1(student: cons) nopause nogroup
```

```
MLwiN 2.26 multilevel model          Number of obs      =      4059
Normal response model
Estimation algorithm: IGLS
Run time (seconds) =          1.62
```

```

Number of iterations =          4
Log likelihood       = -4662.7134
Deviance            =  9325.4268

```

normexam	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
cons	-.1681502	.0539988	-3.11	0.002	-.273986	-.0623144
standlrt	.5599642	.0124436	45.00	0.000	.5355752	.5843531
girl	.1672281	.0340818	4.91	0.000	.100429	.2340273
boysch	.1776197	.1107521	1.60	0.109	-.0394504	.3946897
girlsch	.1589596	.0872538	1.82	0.068	-.0120547	.3299739

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
Level 2: school				
var(cons)	.0811055	.0161794	.0493945	.1128165
Level 1: student				
var(cons)	.5622733	.012581	.5376149	.5869316

The response is normalized and so the fixed part parameters are scaled in standard deviation units. Age 11 scores are standardized and so a one standard deviation increase in age 11 score is associated with a significant 0.560 standard deviation increase in age 16 score, holding all other variables constant ($z = 45.00$, $p < 0.001$). Having adjusted for age 11 scores, girls in mixed sex schools score 0.167 standard deviations higher than boys in mixed sex schools. Boys in boys schools score 0.178 standard deviations higher than boys in mixed sex schools while girls in girls' schools score 0.159 standard deviations higher than girls in mixed sex schools. Neither single-sex school effects is significant. A Wald test confirms that the two effects are also not significantly different from one another ($\chi^2_1 = 0.02$, $p = 0.880$). All the functionality of the **MLwiN Intervals and tests** window is contained within this **test** command.

```

. test boysch = girlsch

( 1) [FP1]boysch - [FP1]girlsch = 0

      chi2( 1) =    0.02
      Prob > chi2 =    0.8796

```

These results suggest that we might constrain the boys' school and girls' school effects to be the same and we could do this by specifying the **constraints()** option. (It is not possible to specify constraints when using Stata's **xtmixed**, **xtmelogit** and **xtmepoisson** commands.) Equivalently, we can remove the boys' and girls' school dummy variables and replace them with a single-sex school dummy variable. Either way, the single-sex school effect is estimated (results not shown) to be 0.165 and significant ($z = 2.17$, $p = 0.030$). Thus, students in single-sex schools score 0.165 standard deviations higher than students of the same sex in mixed sex-schools.

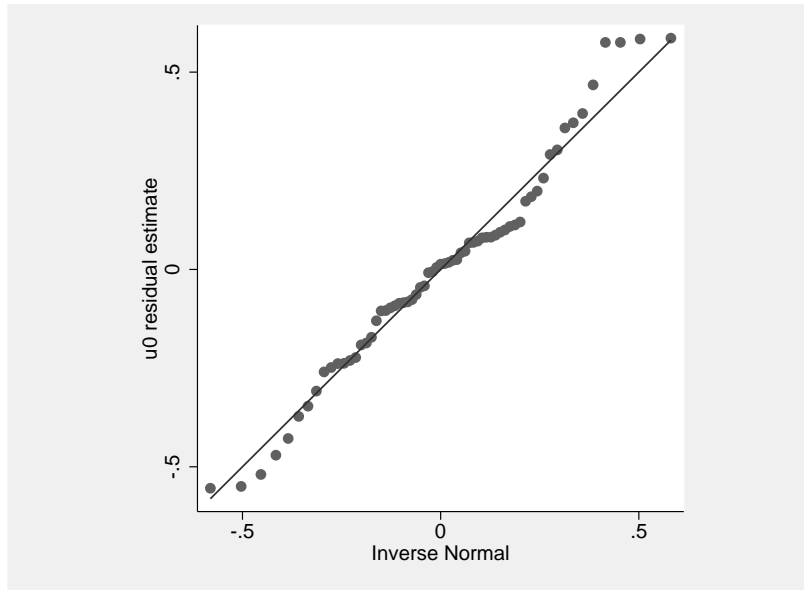


Figure 3: Quantile-quantile plot of the school effects plotted in Stata.

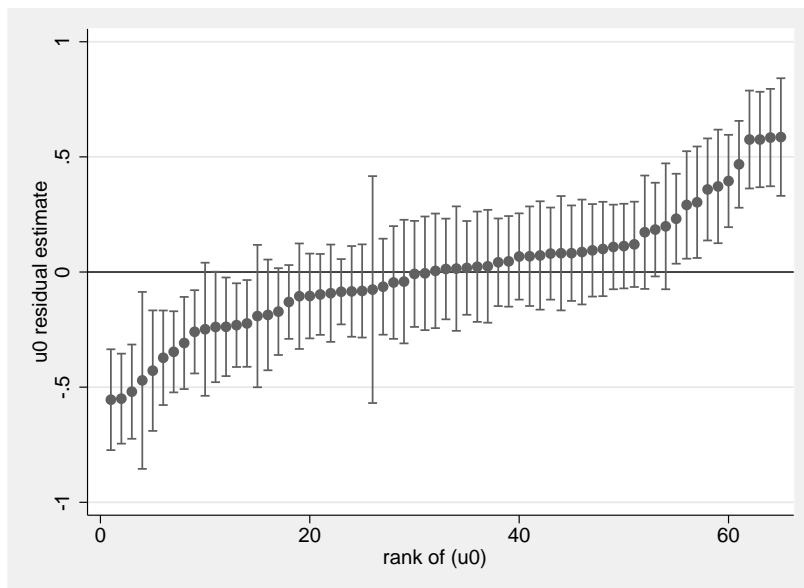


Figure 4: Caterpillar plot of the school effects plotted in Stata.

When we ran the above `runmlwin` command, two new variables, `u0` and `u0se`, appeared in the Stata `Variables` window. These store the empirical Bayes estimates and standard errors of the school random effects, respectively. While there are only 65 schools, these two variables have 4,059 observations, one for each student in the data. The estimate and standard error for each school is simply replicated across all the students within each school. In what follows, we want to base our analysis on just one observation per school and so we create a dummy variable to pick one observation per school at random.

```
. egen pickone_school = tag(school)
```

Next we use the `qnorm` graph command to produce quantile-quantile plots to check whether the random effects are normally distributed. If the random effects are normally distributed, all the data should be plotted along the 45 degree line. We use the `aspectratio(1)` option to specify a one-to-one relationship between the height and the width of the graph's plot region. If desired, further options can be added to alter the axes titles and the spacing of the ticks on the axes. All the graphical functionality of the **MLwiN Residuals** window can be replicated using Stata's many graph commands.

```
. qnorm u0 if pickone_school==1, aspectratio(1)
```

The resulting graph is shown in Figure 3. While the 65 schools do not all lie on the line, they all lie close to the line suggesting that the predicted effects are approximately normally distributed.

Next we shall produce a 'caterpillar plot' of the school effects. The school effects are plotted in ascending rank order and presented with 95% confidence intervals so we can examine those schools which differ significantly from the average school. First we use the `egen` command and the `rank()` function to rank the school effects by the magnitude of their predicted effects.

```
. egen u0rank = rank(u0) if pickone_school==1
```

We then use the `serrbar` graph command to produce the caterpillar plot. The first variable after the command must contain the predicted effects, the second the associated standard errors and the third the rank of the predicted effects. We use the `scale(1.96)` option to obtain 95% confidence limits and the `yline(0)` option to plot a horizontal line at zero which represents the average school in the data. Again, many further options can be added to improve the look and feel of this graph.

```
. serrbar u0 u0se u0rank if pickone_school==1, scale(1.96) yline(0)
```

The resulting graph is shown in Figure 4. The plot shows that approximately two-thirds of schools cannot be distinguished from the overall average.

Using Stata's `count` command we see that 12 schools are significantly less effective than average, while 11 schools are significantly more effective than average.

```
. count if pickone_school==1 & u0 + 1.96*u0se < 0
```

12

```
. count if pickone_school==1 & u0 - 1.96*u0se > 0
```

11

We finish our analysis of this model by dropping `u0` and `u0se` from the data and then storing the model results, naming them as `model2`, as we shall refer back to them later.

```
. drop u0 u0se
. estimates store model2
```

2.3. Random slope model

Model 3 is a two-level random slope model where we allow the coefficient of `standlrtij` to vary randomly across schools. Thus, we allow the relationship between age 11 and age 16 scores to be potentially stronger in some schools than others.

$$\text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + \beta_3 \text{boysch}_j + \beta_4 \text{girlsch}_j + u_{0j} + u_{1j} \text{standlrt}_{ij} + e_{ij}$$

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u0}^2 & \\ \sigma_{u01} & \sigma_{u1}^2 \end{pmatrix} \right\}$$

$$e_{ij} \sim N(0, \sigma_e^2)$$

In the level 2 random part of the model, σ_{u0}^2 measures the variability in schools' intercepts, σ_{u1}^2 measures the variability in schools' slopes and σ_{u01} measures the extent to which schools' intercepts and slopes covary. The correlation between schools' intercepts and slopes can be calculated as $\sigma_{u01} / \sqrt{\sigma_{u0}^2 \sigma_{u1}^2}$.

To fit this model using `runmlwin`, we simply add `standlrt` to the list of variables made random at level 2

```
. runmlwin normexam cons standlrt girl boysch girlsch, ///
> level2(school: cons standlrt, residuals(u)) ///
> level1(student: cons) nopause nogroup
```

```
MLwiN 2.26 multilevel model                Number of obs      =      4059
Normal response model
Estimation algorithm: IGLS
Run time (seconds) =          1.56
Number of iterations =           4
Log likelihood      = -4640.5601
Deviance           =  9281.1201
```

normexam	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
cons	-.1888196	.0513513	-3.68	0.000	-.2894662	-.0881729
standlrt	.554426	.019938	27.81	0.000	.5153482	.5935038
girl	.1682632	.0338217	4.98	0.000	.1019739	.2345525
boysch	.1798662	.0991404	1.81	0.070	-.0144454	.3741778
girlsch	.1748232	.0787633	2.22	0.026	.02045	.3291965

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
Level 2: school				
var(cons)	.0795418	.0159664	.0482482	.1108353
cov(cons,standlrt)	.0201009	.0064875	.0073856	.0328162
var(standlrt)	.0146521	.0044111	.0060065	.0232977

```

Level 1: student |
var(cons) | .5501983 .0124031 .5258886 .574508
-----

```

The fixed part of the model adjusts for the same set of predictors as in Model 2 and, as expected, leads to very similar parameter estimates.

In the random part of the model at level 2 we see that the intercept variance is estimated to be 0.080 while the slope variance is estimated to be 0.015. We will plot graphs to aid our interpretation of these estimates below. Using the `display` command to refer to the parameter estimates by their internal `runmlwin` names, we calculate the correlation between intercept and slopes to be 0.589

```

. display [RP2]cov(cons\standlrt)/sqrt([RP2]var(cons)*[RP2]var(standlrt))

.58879988

```

We can use the `estimates table` command to display the Model 2 and Model 3 parameter estimates side by side. We use the `stats(N deviance ll)` option to additionally display the sample sizes, deviances and log-likelihoods. The remaining three options specify the display precision of the parameter estimates and summary statistics and the width of the Variable column, respectively.

```

. estimates store model3
. estimates table model2 model3, ///
> stats(N deviance ll) b(%4.3f) stfmt(%4.0f) varwidth(18)

```

```

-----
Variable | model2   model3
-----+-----
FP1      |
   cons | -0.168   -0.189
  standlrt | 0.560    0.554
   girl | 0.167    0.168
  boysch | 0.178    0.180
  girlsch | 0.159    0.175
-----+-----
RP2      |
  var(cons) | 0.081    0.080
cov(cons\standlrt) |           0.020
  var(standlrt) |           0.015
-----+-----
RP1      |
  var(cons) | 0.562    0.550
-----+-----
Statistics |
   N | 4059    4059
 deviance | 9325    9281
   ll | -4663   -4641
-----

```

We can use the `lrtest` command to perform a likelihood ratio test to compare this model to the previous random intercept model. The test shows that allowing the effect of age 11 scores to vary across schools leads to a significant improvement in model fit ($\chi^2_2 = 44.31$, $p < 0.001$). Thus, in some schools the relationship between age 11 and age 16 scores is indeed significantly stronger than it is in others schools.

```
. lrtest model2 model3
```

Likelihood-ratio test	LR chi2(2) =	44.31
(Assumption: model2 nested in model3)	Prob > chi2 =	0.0000

To get a better sense of the variability in the school intercepts and slopes we can use the `generate` command, referring once again to the parameter estimates by their internal `runmlwin` names, to calculate the predicted age 16 score for each student based only on their age 11 scores. All the functionality of the **MLwiN Predictions** window can be replicated in this way.

$$\hat{\beta}_0 + \hat{\beta}_1 \text{standlrt}_{ij} + \hat{u}_{0j} + \hat{u}_{1j} \text{standlrt}_{ij}$$

```
. generate ypred = [FP1]cons + [FP1]standlrt*standlrt + u0 + u1*standlrt
```

We then plot these predicted age 16 scores against students' age 11 scores to graphically illustrate the extent to which schools differ from one another. All the functionality of the **MLwiN Customized Graphs** window can be replicated using Stata's many graph commands.

```
. sort school standlrt
. line ypred standlrt, connect(a)
```

The resulting graph is shown in Figure 5. The school lines fan outwards and so the differences between schools appear more pronounced for students with high age 11 scores than they do for students with low age 11 scores.

The exact relationship for how the between school variance changes as a function of age 11 scores is given by the level 2 variance function.

$$\text{var}(u_{0j} + u_{1j} \text{standlrt}_{ij}) = \sigma_{u0}^2 + 2\sigma_{u01} \text{standlrt}_{ij} + \sigma_{u1}^2 \text{standlrt}_{ij}^2$$

We can use the `generate` command to calculate the predicted between school variance for each student. All the functionality of the **MLwiN Variance** window can be replicated in this way.

```
. generate lev2var = [RP2]var(cons) + 2*[RP2]cov(cons\standlrt)*standlrt ///
> + [RP2]var(standlrt)*standlrt^2
```

Plotting these predictions against students' age 11 scores clearly shows that the between school variance increases with age 11 scores (Figure 6).

```
. line lev2var standlrt, sort
```

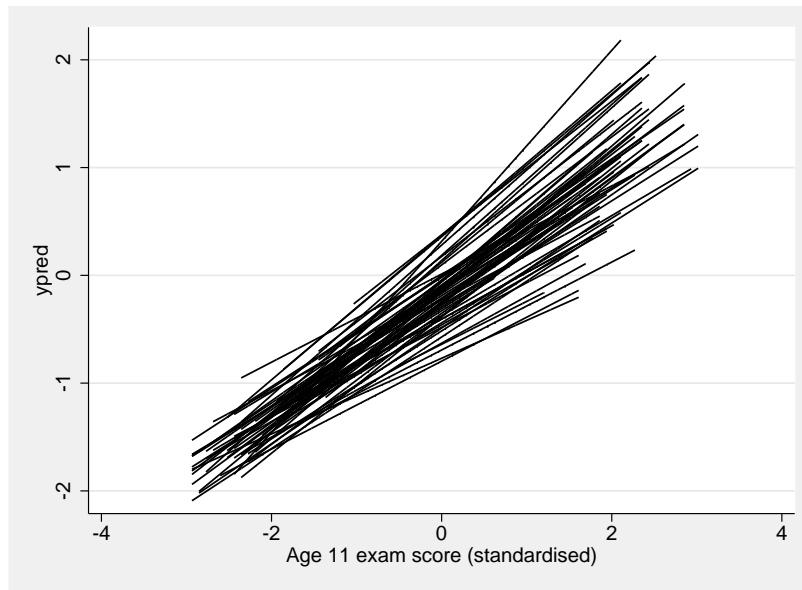


Figure 5: Predicted school lines plotted in Stata.

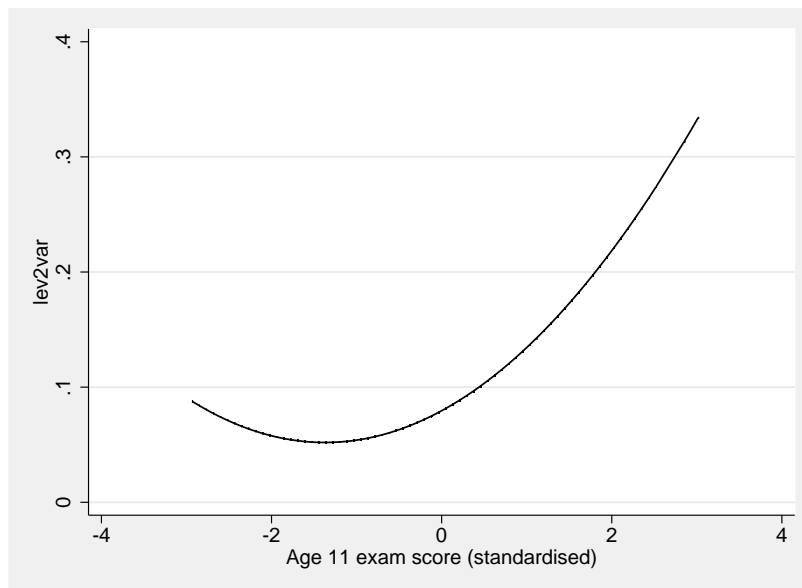


Figure 6: School-level variance function plotted in Stata.

In this section we have focused on using `runmlwin` to fit two-level models for continuous responses. As with `xtmixed`, `runmlwin` can also fit continuous response multilevel models with three or more levels, cross-classifications and sampling weights. However, unlike `xtmixed`, `runmlwin` can additionally fit continuous response multilevel models with multivariate responses, multiple membership structures, complex level 1 variance functions (heteroskedasticity), and linear constraints.

3. Binary response models

In this section we consider how to fit two-level random intercept and random slope models to binary response variables. We use the Bangladeshi contraceptive use data analyzed in Chapter 9 of the **MLwiN** User Manual and Chapter 10 of the **MLwiN** MCMC Manual. Before we describe these data, we first discuss estimation of discrete response multilevel models. The likelihood of the observed data in discrete response multilevel models does not generally have a closed-form expression and so approximate methods of estimation must be used. **MLwiN** offers two approximate methods: quasi-likelihood and MCMC methods.

The idea behind quasi-likelihood methods is to approximate discrete response multilevel models as continuous response multilevel models so that the standard IGLS algorithm can be applied. **MLwiN** provides four quasi-likelihood methods: first and second order marginal quasi-likelihood (MQL1, MQL2) and first and second order penalized quasi-likelihood (PQL1, PQL2). All four methods have been shown to be biased, particularly if sample sizes within level 2 units are small or the response proportion is extreme. These methods should therefore only be used for model exploration; any final model should be fitted by MCMC. PQL2 is the most accurate of the four quasi-likelihood methods, but the least stable and the slowest to converge while MQL1 is the least accurate, but the most stable and fastest to converge. Thus, when fitting exploratory models, **MLwiN** users will often first fit their models by MQL1 and then use these estimates as starting values for running additional iterations by PQL2. The **MLwiN** User Manual provides an accessible introduction to quasi-likelihood methods, explained in the context of the **MLwiN** software. Technical details are given in Goldstein (2011).

MCMC is a simulation approach. After assigning starting values (e.g., the quasi-likelihood estimates) and prior distributions for the model parameters, a Markov chain is used to sequentially sample subsets of parameters from their conditional posterior distributions given current values of the other parameters. (We do not simulate directly from the joint posterior as it is generally intractable.) After an initial burn-in period, when the chain converges to its stationary distribution, the chain is run for a further monitoring period. The means and standard deviations of the sampled parameters from the monitoring period are used as parameter estimates and standard errors while the 2.5th and 97.5th quantiles of these chains provide Bayesian 95% credible intervals, analogous to 95% confidence intervals. When vague or diffuse prior distributions are specified (the default in **MLwiN**), the parameter estimates are essentially maximum likelihood estimates. The **MLwiN** MCMC Manual provides an accessible introduction to MCMC methods, in the context of the **MLwiN** software. Technical details are given in Goldstein (2011).

The data are a sub-sample from the 1989 Bangladesh Fertility Survey (Huq and Cleland 1990). The binary response variable that we consider refers to whether a woman was using contraception at the time of the survey. The full sample was analyzed in Amin, Diamond, and Steele (1997), but with a four-category response that distinguishes between different types of contraceptive method. We shall present ordinal and nominal models for this response in Section 4. The aim of this analysis is to identify factors associated with use of contraception and to examine the extent of between-district variation in contraceptive use. The data have a two-level hierarchical structure, with 2,867 women (level 1) nested within 60 districts (level 2).

```
. use "http://www.bristol.ac.uk/cmm/media/runmlwin/bang.dta", clear
```

The data contain the following variables

- **woman**: woman identifier;
- **district**: district identifier;
- **use**: woman's contraception use (0: no contraception; 1: contraception);
- **use4**: woman's contraceptive use status and method (1: male or female sterilization; 2: modern reversible method; 3: traditional method; 4: no contraception);
- **lc**: number of living children (0: none; 1: one; 2: two; 3: three or more);
- **age**: woman's age (in years), centered on the sample mean of 30 years;
- **urban**: woman's area of residence (0: rural; 1: urban);
- **educ**: woman's level of education (1: none; 2: lower primary; 3: upper primary; 4: secondary);
- **hindu**: woman's religion (0: Muslim; 1: Hindu);
- **dlit**: proportion of women in district who are literate;
- **dpray**: proportion of Muslim women in district who pray daily (a measure of religiosity);
- **cons**: equal to the number 1 for each woman and is used as the intercept term in the models.

3.1. Variance components model

The two-level women-within-districts variance components model for these data is written as

$$\begin{aligned} \text{use}_{ij} &\sim \text{Bernoulli}(\pi_{ij}) \\ \text{logit}(\pi_{ij}) &= \beta_0 + u_j \\ u_j &\sim \text{N}(0, \sigma_u^2) \end{aligned}$$

where use_{ij} is the binary response for whether woman i in district j uses contraception, β_0 is the intercept measuring the log-odds of using contraception in the average district and u_j is a district level random effect assumed normally distributed with zero mean and a constant variance σ_u^2 . No woman-level residual error appears in the linear predictor.

We can fit this model using `runmlwin` and the `discrete()` option where we specify the `distribution(binomial)`, `link(logit)` and `denominator(cons)` sub options to request a binomial response model with a logit link function and a denominator (number of binomial trials) equal to one for each observation. We specify the `pql2` sub option to fit the model by PQL2 as opposed to default estimation by MQL1. In contrast to continuous response models, we do not make `cons` random at level 1 as the model does not include a woman-level residual error.

```
. runmlwin use cons, ///
>   level2(district: cons) ///
>   level1(woman:) ///
>   discrete(distribution(binomial) link(logit) denominator(cons) pql2) ///
>   nopause
```

```

MLwiN 2.26 multilevel model           Number of obs   =   2867
Binomial logit response model
Estimation algorithm: IGLS, PQL2

```

```

-----
                |   No. of      Observations per Group
Level Variable |   Groups      Minimum   Average   Maximum
-----+-----
      district |         60         3      47.8     173
-----

```

```

Run time (seconds) =      1.53
Number of iterations =      5

```

```

-----
      use |      Coef.   Std. Err.      z    P>|z|      [95% Conf. Interval]
-----+-----
      cons |  -0.5101292  .0808406    -6.31  0.000    -0.6685739   -0.3516845
-----

```

```

-----
Random-effects Parameters |   Estimate   Std. Err.      [95% Conf. Interval]
-----+-----
Level 2: district
      var(cons) |   .2650954   .0703541     .1272039     .4029869
-----

```

The formatting of the model output is similar to that for continuous response models and again follows that of other *Stata* estimation commands. The top left and right blocks of output report that **MLwiN** 2.26 was used to fit a binomial logit response model to an estimation sample of 2,867 observations (women) using the IGLS algorithm and PQL2. The first table of output informs us that the model is fitted to 60 districts where the number of women per district ranges from 3 to 173 with an average of 47.8 women. The next block of output reports that `runmlwin` took 1.53 seconds and 5 iterations of the IGLS algorithm to fit the model. Importantly, no log-likelihood or deviance statistic is reported as the model is fitted by quasi-likelihood methods rather than by maximum likelihood.

As quasi-likelihood estimates are known to be biased, we refit the model by MCMC by simply adding the `mcmc(on)` option to the previous `runmlwin` command. **MLwiN** will then fit the model using its default MCMC estimation settings: vague prior distributions, a burn-in period of 500 iterations followed by a monitoring period of 5,000 iterations. (`runmlwin` options are available to alter all of these estimation settings.) Initial values must always be provided for MCMC estimation. The `initsprevious` option is specified in order to provide the PQL2 parameter estimates from the previous model as starting values for the current model.

```

. runmlwin use cons, ///
>   level2(district: cons) ///
>   level1(woman:) ///
>   discrete(distribution(binomial) link(logit) denom(cons)) ///
>   mcmc(on) initsprevious nopause nogroup

```

```

MLwiN 2.26 multilevel model                Number of obs      =      2867
Binomial logit response model
Estimation algorithm: MCMC
Burnin                                     =          500
Chain                                       =          5000
Thinning                                    =           1
Run time (seconds)                         =          12.7
Deviance (dbar)                            =        3678.37
Deviance (thetabar)                       =        3636.22
Effective no. of pars (pd) =              42.15
Bayesian DIC                               =        3720.52

```

```

-----
      use |      Mean   Std. Dev.   ESS   P       [95% Cred. Interval]
-----+-----
      cons |  -0.5078102  .0870904   133  0.000  -0.6763612  -0.335745
-----

```

```

-----
Random-effects Parameters |      Mean   Std. Dev.   ESS   [95% Cred. Int]
-----+-----
Level 2: district        |
      var(cons) |  .2853694  .0808969   704  .1547615  .4686836
-----

```

The top left block of output states that it took 12.7 seconds to run the model for the default burn-in period of 500 iterations and monitoring period of 5,000 iterations.

In the fixed part parameters table, the first and second columns of numbers present the means (parameter estimates) and standard deviations (standard errors) of the parameter posterior distributions. In contrast to model output after fitting by IGLS, the third and fourth columns do not present the usual z ratios and p values as MCMC does not assume that the parameters follow asymptotic normal sampling distributions. Instead, column three presents the effective sample size (ESS) for each parameter, an estimate of the equivalent number of independent iterations that the chain represents. The ESS will typically be less than the number of actual iterations because the chains are positively autocorrelated (they are Markov chains). Column four presents one-tailed p values based on the posterior distributions. For a positive estimate, the p value is the proportion of that parameter's posterior distribution that is below zero. For a negative estimate, the p value is the proportion of that parameter's posterior distribution that is above zero. The fifth and sixth columns give the 2.5th and 97.5th quantiles of the posterior distribution, resulting in 95% Bayesian credible intervals.

The intercept β_0 is estimated to be -0.508 and is interpreted as the log-odds of using contraception in the average district. The corresponding odds and probability of using contraception are derived as $\exp(\beta_0)$ and $\exp(\beta_0)/\{1 + \exp(\beta_0)\}$ and are estimated to be 0.602 and 0.375, respectively. Thus, almost 40% of the women use some method of contraception.

```
. display exp([FP1]cons)
```

```
.60181196
```

```
. display exp([FP1]cons)/(1 + exp([FP1]cons))
```

```
.375707
```

The ESS is 133, implying that the sample of 5,000 values is only equivalent to only 133 independent iterations. We might therefore choose to refit the model specifying a longer burn-in and monitoring period.

In the random part parameters table, the between-district variance σ_u^2 is estimated to be 0.285. The ESS for this parameter is 704 and so its parameter chain is less autocorrelated than the intercept's parameter chain. In Section 1, we discussed the ICC and VPC statistics for measuring clustering or dependence in continuous response data. In the case of binary and other discrete response models, there is no single ICC or VPC value as the level 1 variance is a function of the mean. A popular solution is to formulate the model in terms of a continuous latent response variable which underlies the observed binary response. It can then be shown that the ICC and VPC statistics, in terms of the underlying latent response, are calculated as $\sigma_u^2/(\sigma_u^2 + \pi^2/3)$. In our case, the latent variable underlying women's observed behavior would be interpreted as their propensity to use contraception.

```
. display [RP2]var(cons)/([RP2]var(cons) + (_pi^2)/3)
```

```
.0798183
```

Thus, the expected correlation in the propensity to use contraception between two women from the same district is just 0.080. Interpreted as a VPC we would say that 8% of the variation in women's propensity to use contraception lies between districts.

In addition to `runmlwin`, we have also developed a second command `mcmcsum` (distributed with `runmlwin`) to calculate more detailed graphical and statistical MCMC diagnostics for the parameter chains. The `mcmcsum` command replicates all the functionality of the **MLwiN Trajectories** window. For example, specifying the `mcmcsum` command's `trajectories` option produces trajectory plots (trace plots) of the deviance statistic and each model parameter. The resulting graph is shown in Figure 7.

```
. mcmcsum, trajectories
```

We can examine a specific parameter chain in detail by listing it after the `mcmcsum` command and then specifying the `fiveway` option to produce a five-way MCMC graphical diagnostic plot. We do this below for the between-district variance σ_u^2 .

```
. mcmcsum [RP2]var(cons), fiveway
```

The resulting graph is shown in Figure 8. On the first row, the left panel plots the trace of the chain while the right panel plots a smoothed histogram of the chain. The smoothed histogram shows the posterior distribution of the between-district variance to be positively skewed. On the second row, the left panel is an auto-correlation function (ACF) plot showing the autocorrelation between iteration t and $t - k$ while the right panel is a partial auto-correlation function (PACF) showing the autocorrelation between iteration t and $t - k$, having accounted for iterations $t - 1, \dots, t - (k - 1)$. The less correlated the chain the better. Here the chain looks moderately correlated and we may wish to run the chain for longer. The

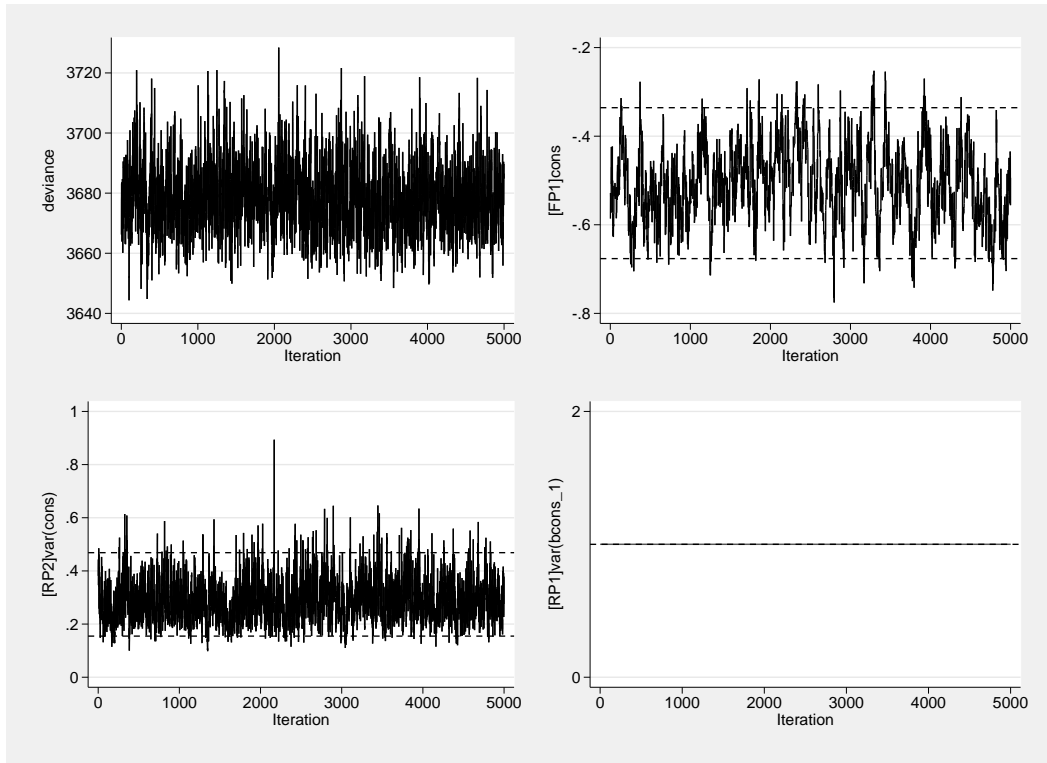


Figure 7: Deviance and parameter chains plotted in Stata.

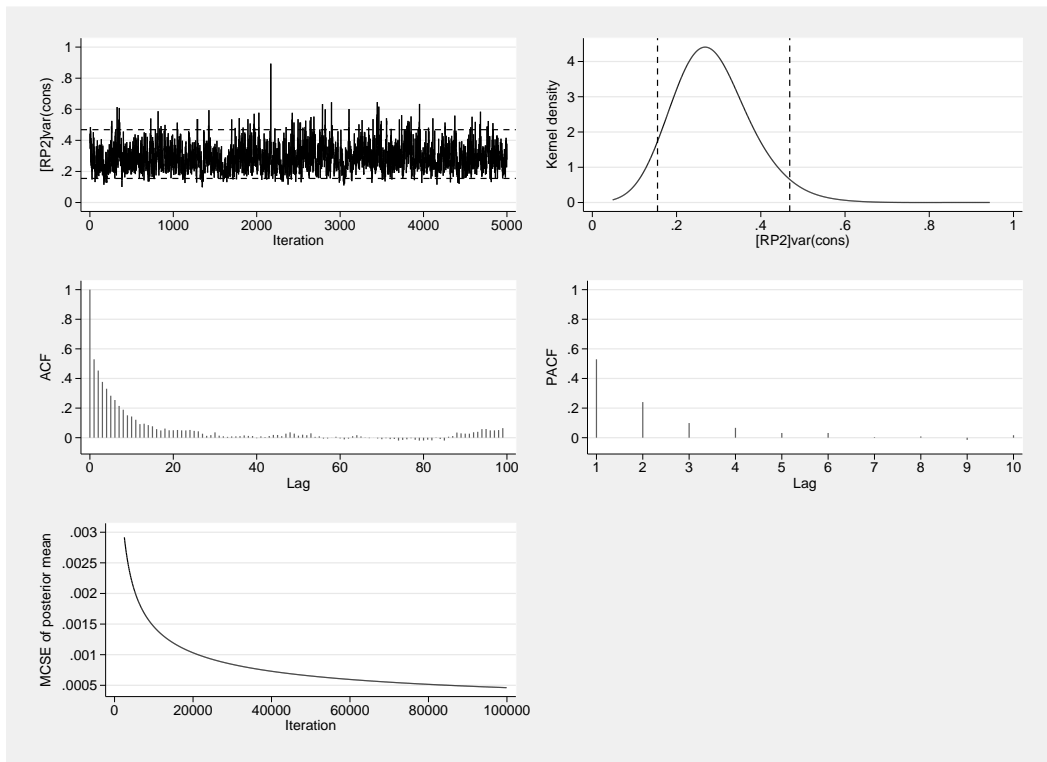


Figure 8: Five-way MCMC graphical diagnostics plot plotted in Stata.

single graph on the third row plots the Monte Carlo standard error (MCSE), an indication of how much error is in the mean estimate due to the fact that MCMC is used. As the number of iterations increases the MCSE tends to 0. The MCSE can be used to calculate how long to run the chain to achieve a mean estimate with a particular desired MCSE. Full details are provided in the **MLwiN** MCMC Manual.

Specifying the `mcmcsum` command for a specific parameter with no options gives an expanded set of summary statistics for that parameter. These include the mode, median and other quantiles of the posterior distribution along with the Brooks-Draper and Raftery-Lewis statistics for helping to judge burn-in length. Full details are provided in the **MLwiN** MCMC Manual.

```
. mcmcsum [RP2]var(cons)
```

[RP2] var(cons)					

		Percentiles			
Mean	.2853694	0.5%	.1298765	Thinned Chain Length	5000
MCSE of Mean	.0020629	2.5%	.1547615	Effective Sample Size	704
Std. Dev.	.0808969	5%	.1700509	Raftery Lewis (2.5%)	12246
Mode	.2678715	25%	.2277897	Raftery Lewis (97.5%)	7558
P(mean)	0			Brooks Draper (mean)	3271
P(mode)	0	50%	.2773703		
P(median)	0				
		75%	.3309639		
		95%	.4328758		
		97.5%	.4686836		
		99.5%	.5472924		

Finally, it is worth comparing the PQL2 and MCMC estimates. For this particular model and dataset the differences between the PQL2 and MCMC results are in fact rather small. Recall that quasi-likelihood estimates become more biased the smaller sample sizes are within level 2 units and the more extreme the response proportion. In our case, we have both large level 2 units (there are on average 48 women per district) and a non-extreme overall response proportion (the probability of using contraception in the average district is estimated to be 0.376).

3.2. Random intercept model with predictor variables

In our next model we adjust for woman's age, number of living children, whether the woman lives in an urban or rural area of residence, education status and religion.

$$\text{use}_{ij} \sim \text{Bernoulli}(\pi_{ij})$$

$$\begin{aligned} \text{logit}(\pi_{ij}) = & \beta_0 + \beta_1 \text{age}_{ij} + \beta_2 \text{lc1}_{ij} + \beta_3 \text{lc2}_{ij} + \beta_4 \text{lc3plus}_{ij} + \beta_5 \text{urban}_{ij} \\ & + \beta_6 \text{edlprim}_{ij} + \beta_7 \text{eduprim}_{ij} + \beta_8 \text{edsecplus}_{ij} + \beta_9 \text{hindu}_{ij} + u_j \end{aligned}$$

$$u_j \sim N(0, \sigma_u^2)$$

The number of living children and women's education levels are both entered into the model as series of dummy variables. We start by generating these dummy variables

```
. generate lc1 = (lc==1)
. generate lc2 = (lc==2)
. generate lc3plus = (lc==3)
. generate edlprim = (educ==2)
. generate eduprim = (educ==3)
. generate edsecprim = (educ==4)
```

Next, we fit the model by MCMC, where we first fit the model by PQL2 to obtain sensible starting values for the model parameters. We prefix the `runmlwin` command with the `quietly` command to suppress the PQL2 model output. In the MCMC model, we specify the `burnin(1000)` and `chain(10000)` sub options within `mcmc()` to illustrate how to manually specify the burn-in and monitoring period.

```
. quietly runmlwin use cons age lc1 lc2 lc3plus urban ///
> edlprim eduprim edsecplus hindu, ///
> level2(district: cons) ///
> level1(woman:) ///
> discrete(distribution(binomial) link(logit) denom(cons) pql2) ///
> nopause
.
. runmlwin use cons age lc1 lc2 lc3plus urban ///
> edlprim eduprim edsecplus hindu, ///
> level2(district: cons, residuals(u)) ///
> level1(woman:) ///
> discrete(distribution(binomial) link(logit) denom(cons)) ///
> mcmc(burnin(1000) chain(10000)) initsprevious nopause nogroup
```

```
MLwiN 2.26 multilevel model                Number of obs      =      2867
Binomial logit response model
Estimation algorithm: MCMC
Burnin                      =      1000
Chain                        =     10000
Thinning                     =         1
Run time (seconds)          =         72
Deviance (dbar)             =     3391.73
Deviance (thetabar)         =     3343.66
Effective no. of pars (pd) =         48.07
Bayesian DIC                 =     3439.79
```

use	Mean	Std. Dev.	ESS	P	[95% Cred. Interval]	
cons	-2.058425	.1369046	160	0.000	-2.332103	-1.788479
age	-.0170395	.0065595	501	0.005	-.029798	-.0042716
lc1	1.147392	.1319565	486	0.000	.8803015	1.410263
lc2	1.5051	.1436951	309	0.000	1.223534	1.78126
lc3plus	1.497378	.1421594	207	0.000	1.220985	1.770245
urban	.5231315	.1058621	1019	0.000	.3170539	.7259818
edlprim	.2506106	.1325655	1617	0.029	-.0066221	.5117957
eduprim	.7319607	.1434418	1799	0.000	.4501632	1.015892

edsecplus		1.18517	.1267956	1292	0.000	.9327023	1.436409
hindu		.4295443	.126547	1621	0.000	.1771066	.6758361

Random-effects Parameters		Mean	Std. Dev.	ESS		[95% Cred. Int]	

Level 2: district							
		var(cons)	.2576323	.0762447	1163	.1383612	.4352944

The age effect is negative and its 95% credible interval does not include zero implying a significant negative effect of age; older women are less likely to use contraception all else being equal. All three living children dummy variables are positive and their 95% credible intervals similarly exclude zero and so women with children appear significantly more likely to use contraception than childless women, particularly those women with two or three children. Urban also has a significant positive effect and so women living in urban areas appear more likely to use contraception than similar women living in rural areas. The education dummy variables are positive and increase in magnitude with increasing levels of education suggesting that the more educated women are the more likely they are to use contraception. Finally, the Hindu effect is positive and significant and so Hindu women are more likely to use contraception than otherwise equivalent Muslim women.

All parameters are measured on the log-odds (logit) scale. We can redisplay them as odds ratios by rerunning the `runmlwin` command with only the `or` option. We additionally specify the `noheader` and `noretable` options to suppress the header and also the random effects table from also being redisplayed.

```
. runmlwin, noheader noretable or
```

use		Odds Ratio	Std. Dev.	ESS	P	[95% Cred. Interval]	

cons		.128853	.0176396	165	0.000	.0970913	.1672143
age		.983126	.0064464	500	0.005	.9706416	.9957376
lc1		3.177508	.4211868	495	0.000	2.411627	4.097032
lc2		4.551389	.6584419	310	0.000	3.39918	5.937334
lc3plus		4.515563	.6501625	207	0.000	3.390527	5.872294
urban		1.696776	.1798448	1028	0.000	1.373077	2.066759
edlprim		1.296161	.1728697	1612	0.029	.9933997	1.668284
eduprim		2.100653	.3028464	1794	0.000	1.568568	2.761826
edsecplus		3.297643	.4197159	1288	0.000	2.541367	4.205568
hindu		1.548892	.1963568	1635	0.000	1.193758	1.965676

For example, women living in urban areas are 1.697 times more likely to use contraception than similar women living in rural areas. The standard deviations, ESS and credible intervals are also all rescaled. For example, the 95% credible interval for this urban odds ratio is (1.373, 2.067).

In the random part of the model the between-district variance is now estimated to be 0.258 compared to 0.285 in the variance components model. The ICC and VPC are therefore now lower than they were previously suggesting that even though all the predictors are level 1 variables, they have explained away relatively more variation between districts than within districts. There is clearly substantial district variation in women's background characteristics across the sample.

3.3. Random slope models

In our next model we introduce a random coefficient to allow the magnitude of the urban-rural differential to vary across districts.

$$\begin{aligned} \text{use}_{ij} &\sim \text{Bernoulli}(\pi_{ij}) \\ \text{logit}(\pi_{ij}) &= \beta_0 + \beta_1 \text{age}_{ij} + \beta_2 \text{lc1}_{ij} + \beta_3 \text{lc2}_{ij} + \beta_4 \text{lc3plus}_{ij} + \beta_5 \text{urban}_{ij} \\ &\quad + \beta_6 \text{edlprim}_{ij} + \beta_7 \text{eduprim}_{ij} + \beta_8 \text{edsecplus}_{ij} + \beta_9 \text{hindu}_{ij} + u_{0k} + u_{5k} \\ \begin{pmatrix} u_{0k} \\ u_{5k} \end{pmatrix} &\sim \text{N} \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u0}^2 & \\ & \sigma_{u5}^2 \end{pmatrix} \right\} \end{aligned}$$

To fit this model using `runmlwin`, we simply add `urban` to the list of variables made random at level 2. We specify `orthogonal` within `mcmc()` to reparameterize the model using orthogonal fixed effects. This reparameterization tends to make the MCMC chains less autocorrelated and so we can typically run the model for fewer iterations to achieve the same ESS for each parameter. Indeed the ESS for the fixed part parameters are mostly higher than they were in the previous model where we ran the monitoring period for twice as many iterations.

```
. quietly runmlwin use cons age lc1 lc2 lc3plus urban ///
>   edlprim eduprim edsecplus hindu, ///
>   level2(district: cons urban) ///
>   level1(woman:) ///
>   discrete(distribution(binomial) link(logit) denom(cons) pql2) ///
>   nopause
.
. runmlwin use cons age lc1 lc2 lc3plus urban ///
>   edlprim eduprim edsecplus hindu, ///
>   level2(district: cons urban, residuals(u)) ///
>   level1(woman:) ///
>   discrete(distribution(binomial) link(logit) denom(cons)) ///
>   mcmc(orthogonal) initsprevious nopause nogroup
```

MLwiN 2.26 multilevel model	Number of obs	=	2867
Binomial logit response model			
Estimation algorithm: MCMC			
Burnin	=	500	
Chain	=	5000	
Thinning	=	1	
Run time (seconds)	=	45.3	

```

Deviance (dbar)          =    3359.47
Deviance (thetabar)     =    3296.27
Effective no. of pars (pd) =     63.20
Bayesian DIC             =    3422.67

```

use	Mean	Std. Dev.	ESS	P	[95% Cred. Interval]	
cons	-2.093502	.145026	323	0.000	-2.383879	-1.822029
age	-.0186721	.0065416	1175	0.002	-.0317001	-.0058585
lc1	1.165411	.1348019	1022	0.000	.8921169	1.426872
lc2	1.530448	.1514643	974	0.000	1.233069	1.824439
lc3plus	1.53144	.1508309	1135	0.000	1.249121	1.833495
urban	.5722347	.1383094	316	0.000	.2983797	.8484937
edlprim	.2443008	.1245475	1150	0.023	.0061003	.4831241
eduprim	.730011	.146977	1030	0.000	.436473	1.021763
edsecplus	1.179433	.1267397	1128	0.000	.9300272	1.420014
hindu	.5035544	.1301246	706	0.000	.253488	.7734384

Random-effects Parameters	Mean	Std. Dev.	ESS	[95% Cred. Int]	
Level 2: district					
var(cons)	.4004016	.1170398	289	.218859	.6746963
cov(cons,urban)	-.2860598	.1187465	171	-.561482	-.0959287
var(urban)	.3917124	.1653934	146	.1442434	.7695333

The fixed part of the model adjusts for the same set of predictors as before and, as expected, leads to very similar parameter estimates. In the random part of the model, the between district variance is now a function of `urban`

$$\text{var}(u_{0j} + u_{5j}\text{urban}_{ij}) = \sigma_{u0}^2 + 2\sigma_{u05}\text{urban}_{ij} + \sigma_{u5}^2\text{urban}_{ij}^2$$

The residual variance between districts for women living in rural areas (`urbanij = 0`) is estimated by σ_{u0}^2

```
. display [RP2]var(cons)
```

```
.40040159
```

while the residual variance between districts for women living in urban areas (`urbanij = 1`) is $\sigma_{u0}^2 + 2\sigma_{u05} + \sigma_{u5}^2$.

```
. display [RP2]var(cons) + 2*[RP2]cov(cons\urban) + [RP2]var(urban)
```

```
.21999443
```

These results show that there is greater district-level variation in the probability of using contraception in rural areas than in urban areas.

Finally, we will include the two district-level explanatory variables, `dlit` and `dpray`, to see whether they explain some of the district-level variation in urban and rural areas.

$$\text{use}_{ij} \sim \text{Bernoulli}(\pi_{ij})$$

$$\begin{aligned} \text{logit}(\pi_{ij}) = & \beta_0 + \beta_1 \text{age}_{ij} + \beta_2 \text{lc1}_{ij} + \beta_3 \text{lc2}_{ij} + \beta_4 \text{lc3plus}_{ij} + \beta_5 \text{urban}_{ij} \\ & + \beta_6 \text{edlprim}_{ij} + \beta_7 \text{eduprim}_{ij} + \beta_8 \text{edsecplus}_{ij} + \beta_9 \text{hindu}_{ij} \\ & + \beta_{10} \text{dlit}_j + \beta_{11} \text{dpray}_j + u_{0k} + u_{5k} \\ \begin{pmatrix} u_{0k} \\ u_{5k} \end{pmatrix} \sim & N \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u0}^2 & \\ & \sigma_{u5}^2 \end{pmatrix} \right\} \end{aligned}$$

Fitting this model gives the following model results

```
. quietly runmlwin use cons age lc1 lc2 lc3plus urban ///
>   edlprim eduprim edsecplus hindu dlit dpray, ///
>   level2(district: cons urban) ///
>   level1(woman:) ///
>   discrete(distribution(binomial) link(logit) denom(cons) pql2) ///
>   nopause
.
. runmlwin use cons age lc1 lc2 lc3plus urban ///
>   edlprim eduprim edsecplus hindu dlit dpray, ///
>   level2(district: cons urban) ///
>   level1(woman:) ///
>   discrete(distribution(binomial) link(logit) denom(cons)) ///
>   mcmc(orthogonal) initsprevious nopause nogroup
```

```
MLwiN 2.26 multilevel model                Number of obs      =      2867
Binomial logit response model
Estimation algorithm: MCMC
Burnin                      =          500
Chain                        =         5000
Thinning                     =           1
Run time (seconds)          =          51.4
Deviance (dbar)             =        3356.43
Deviance (thetabar)        =        3292.10
Effective no. of pars (pd) =         64.33
Bayesian DIC                 =        3420.76
```

	use	Mean	Std. Dev.	ESS	P	[95% Cred. Interval]	
	cons	-1.735759	.2942317	118	0.000	-2.329606	-1.162001
	age	-.0185978	.0067268	1147	0.003	-.0312961	-.0046087
	lc1	1.182757	.1326589	999	0.000	.9278888	1.440731
	lc2	1.545269	.1500845	924	0.000	1.245595	1.842595

lc3plus		1.547397	.1539206	1028	0.000	1.248998	1.854816
urban		.5263794	.1589417	114	0.001	.2011249	.8388008
edlprim		.2402252	.13064	1177	0.028	-.0114425	.5070141
eduprim		.7391772	.1477399	1114	0.000	.4453486	1.03165
edsecplus		1.20151	.1321474	963	0.000	.9412796	1.457263
hindu		.5010918	.1375902	595	0.000	.235527	.7780737
dlit		2.217206	1.76005	203	0.108	-1.273744	5.675169
dpray		-1.426267	.5541483	164	0.004	-2.505237	-.4128276

Random-effects Parameters		Mean	Std. Dev.	ESS	[95% Cred. Int]
Level 2: district					
var(cons)		.3684094	.1166008	252	.1841991 .6359417
cov(cons,urban)		-.2918086	.1311375	144	-.5992982 -.08096
var(urban)		.4562388	.2106555	111	.1412175 .9463967

The effect of the proportion of literate women in the district has a positive, but non-significant effect on the probability of using contraception. District-level religiosity has a significant negative effect, with women living in districts with higher levels of religiosity being less likely to use contraception. The residual between-district variation is now 0.368 for rural areas and 0.241 for urban areas and so some of district-level variation in rural areas is explained by differences in religiosity, but the variation in urban areas is almost unchanged.

```
. display [RP2]var(cons)
```

```
.36840943
```

```
. display [RP2]var(cons) + 2*[RP2]cov(cons\urban) + [RP2]var(urban)
```

```
.24103108
```

In this section we have focused on using the `runmlwin` command to fit two-level logistic models for binary responses. As with `xtmelogit`, we can extend these models to include three or more levels, cross-classifications, and binomial (proportion) responses. However, unlike `xtmelogit`, `runmlwin` can additionally fit models with different link functions (probit and complementary log-log) and can also fit multivariate binary and binomial response models.

4. More advanced models

In this section we briefly consider how to fit three more advanced multilevel models using `runmlwin`: a multilevel multivariate mixed response type model, a multilevel ordinal response model and a multilevel nominal response model. These models cannot be fitted using Stata's multilevel modeling commands (`xtmixed`, `xtmelogit` and `xtmepoisson`). For simplicity we continue to analyze the education and demography example datasets introduced in the last two sections.

4.1. Multivariate mixed response type model

In this example we fit a multivariate mixed response type model to the London exam scores data. Specifically, we fit a two-level students-within-schools bivariate response model for students' age 11 scores `standlrt` and a dichotomized version of students' age 16 scores `pass` which equals 1 if `normexam` is positive and 0 otherwise. The model is a random intercept model where we only adjust for student gender. We formulate the binary response in terms of an underlying latent response pass_{ij}^* which can be interpreted as the propensity with which student i in school j passes their exam.

$$\begin{aligned} \text{standlrt}_{ij} &= \beta_0^{(1)} + \beta_1^{(1)} \text{girl}_{ij} + u_j^{(1)} + e_{ij}^{(1)} \\ \text{pass}_{ij}^* &= \beta_0^{(2)} + \beta_1^{(2)} \text{girl}_{ij} + u_j^{(2)} + e_{ij}^{(2)} \\ \begin{pmatrix} u_j^{(1)} \\ u_j^{(2)} \end{pmatrix} &\sim N \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u^{(1)}}^2 & \\ \sigma_{u^{(1,2)}} & \sigma_{u^{(2)}}^2 \end{pmatrix} \right\} \\ \begin{pmatrix} e_{ij}^{(1)} \\ e_{ij}^{(2)} \end{pmatrix} &\sim N \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{e^{(1)}}^2 & \\ \sigma_{e^{(1,2)}} & \sigma_{e^{(2)}}^2 \end{pmatrix} \right\} \end{aligned}$$

Since pass_{ij}^* is unobservable, its variance $\sigma_{e^{(2)}}^2$ is not identified, but is implicitly set by choosing the link function (logit: $\sigma_{e^{(2)}}^2 = \pi^2/3$; probit: $\sigma_{e^{(2)}}^2 = 1$). We shall specify the probit link function.

To specify this multivariate mixed response type model using `runmlwin`, we introduce parentheses into the fixed and random parts of the model which include the `equation()` option to declare which variables appear in which response equation. To ease model interpretation, we use the `corr` option to present the level 2 and level 1 covariance parameters as correlations. Fitting this model first by IGLS MQL1 and then by MCMC produces

```
. quietly runmlwin (standlrt cons girl, eq(1)) (pass cons girl, eq(2)), ///
> level2(school: (cons, eq(1)) (cons, eq(2))) ///
> level1(student: (cons, eq(1))) ///
> discrete(distribution(normal binomial) link(probit) denom(cons cons)) ///
> nopause
.
. runmlwin (standlrt cons girl, eq(1)) (pass cons girl, eq(2)), ///
> level2(school: (cons, eq(1)) (cons, eq(2))) ///
> level1(student: (cons, eq(1))) ///
> discrete(distribution(normal binomial) link(probit) denom(cons cons)) ///
> mcmc(orthogonal) initsprevious ///
> nopause nogroup corr
```

```
MLwiN 2.26 multilevel model           Number of obs       =       4059
Multivariate response model
Estimation algorithm: MCMC
Burnin                               =           500
Chain                                 =          5000
Thinning                              =             1
Run time (seconds)                   =           290
```

Deviance (dbar)	=	.					
Deviance (thetabar)	=	.					
Effective no. of pars (pd)	=	.					
Bayesian DIC	=	.					
		Mean	Std. Dev.	ESS	P	[95% Cred. Interval]	

standlrt							
cons_1		-.1185641	.050478	47	0.006	-.2214012	-.0248286
girl_1		.1551671	.0419346	192	0.000	.0746792	.2366822

pass							
cons_2		-.0964004	.0770889	28	0.098	-.2447352	.0519815
girl_2		.1944707	.0536904	182	0.000	.0885649	.2924222

Random-effects Parameters			Mean	Std. Dev.	ESS	[95% Cred. Int]	

Level 2: school							
var(cons_1)		.0976565	.020842	1160	.063758	.14618	
corr(cons_1,cons_2)		.7361008	.0724239	934	.5699684	.8553826	
var(cons_2)		.2071477	.0450404	886	.1338541	.3140524	

Level 1: student							
var(cons_1)		.9012652	.0196933	354	.8633361	.9412403	
corr(cons_1,bcons_2)		.5898498	.0146979	344	.5624947	.6196466	
var(bcons_2)		1	0	0	1	1	

In terms of age 11 scores, we see that girls are predicted to score 0.155 of a standard deviation higher at age 11 than boys. In terms of passing their age 16 exams, girls are again predicted to outperform boys: 54% of girls are predicted to pass their age 16 exams compared to 46% of boys.

```
. display normal([FP2]cons_2)

.46160127

. display normal([FP2]cons_2 + [FP2]girl_2)

.53906177
```

The correlations between students' age 11 scores and their propensity to pass their age 16 exams are 0.736 and 0.590 at the school and student level, respectively. Unsurprisingly, schools with the highest scoring intakes tend to perform best five years later. Similarly, within schools, the higher scoring students at age 11 are the ones which perform best at age 16. These results are consistent with those reported in Section 2.

In the above example, we have fitted a two-level bivariate response model where the fixed and random parts of the model are the same for both response equations. However, we can also use `runmlwin` to specify more complex multilevel multivariate mixed response type models. We can extend the above model to fit three- and higher-level hierarchical data, cross-classified data and response vectors with three or more response variables of potentially varying types (continuous and binary). We can also specify different fixed and random parts in each response equation, allowing certain variables to appear in some equations and not others. We can also specify some parameters to be common (constrained) across equations while others are separately estimated. Random coefficients and variance functions can be added to any equation at any level in the usual way.

4.2. Ordinal response model

In this example we fit an ordinal logistic model to the Bangladeshi contraceptive use data. Specifically, we fit a two-level women-within-districts ordinal response model to the four category contraceptive use variable `use4`. We view the four categories of contraception as being increasingly effective means of birth control: using no contraception (`use4ij = 4`) is the least effective means, then traditional methods (`use4ij = 3`), then modern reversible methods (`use4ij = 2`), and finally sterilization is the most effective means (`use4ij = 1`). The model is a random intercept model where we only adjust for women's age, number of living children and whether she lives in an urban area. We set the fourth category of the response (using no contraception) to be the reference category.

$$\begin{aligned} \text{use4}_{ij} &\sim \text{Multinomial} \left(\pi_{ij}^{(1)}, \pi_{ij}^{(2)}, \pi_{ij}^{(3)} \right) \\ \log \left(\pi_{ij}^{(1)} / \pi_{ij}^{(2,3,4)} \right) &= \beta_0^{(1)} + \beta_1 \text{age}_{ij} + \beta_2 \text{lc1}_{ij} + \beta_3 \text{lc2}_{ij} + \beta_4 \text{lc3plus}_{ij} + \beta_5 \text{urban}_{ij} + u_j \\ \log \left(\pi_{ij}^{(1,2)} / \pi_{ij}^{(3,4)} \right) &= \beta_0^{(2)} + \beta_1 \text{age}_{ij} + \beta_2 \text{lc1}_{ij} + \beta_3 \text{lc2}_{ij} + \beta_4 \text{lc3plus}_{ij} + \beta_5 \text{urban}_{ij} + u_j \\ \log \left(\pi_{ij}^{(1,2,3)} / \pi_{ij}^{(4)} \right) &= \beta_0^{(3)} + \beta_1 \text{age}_{ij} + \beta_2 \text{lc1}_{ij} + \beta_3 \text{lc2}_{ij} + \beta_4 \text{lc3plus}_{ij} + \beta_5 \text{urban}_{ij} + u_j \\ u_j &\sim N \left(0, \sigma_u^2 \right) \end{aligned}$$

The terms $\pi_{ij}^{(1)}$, $\pi_{ij}^{(2)}$ and $\pi_{ij}^{(3)}$ denote the probabilities of using method 1, 2 and 3, respectively, while $\pi_{ij}^{(2,3,4)}$, $\pi_{ij}^{(1,2)}$, $\pi_{ij}^{(3,4)}$ and $\pi_{ij}^{(1,2,3)}$ are different cumulative probabilities. The response has four categories and so the model is set up as three log-odds contrasts. The first log odds contrast models the log odds of using method 1 versus methods 2, 3 or 4. The second log odds contrast models the log odds of using method 1 or 2 versus using methods 3 or 4. The third log odds contrast models the log odds of using methods 1, 2 or 3 versus using method 4. Thus, in each log-odds contrast we contrast more effective approaches to birth control with less effective approaches. Separate intercepts (threshold parameters) are estimated in each log-odds contrast. The effects of all predictor variables are assumed constant across the log-odds contrasts (proportional odds assumption).

Fitting this model first by IGLS MQL1 and then by MCMC produces

```
. quietly runmlwin use4 cons (age lc1 lc2 lc3plus urban, contrast(1/3)), ///
> level2(district: (cons, contrast(1/3))) ///
```



```

> level1(woman:) ///
> discrete(dist(multinomial) link(ologit) denom(cons) basecategory(4)) ///
> nopause
.
. runmlwin use4 cons (age lc1 lc2 lc3plus urban, contrast(1/3)), ///
> level2(district: (cons, contrast(1/3))) ///
> level1(woman:) ///
> discrete(dist(multinomial) link(ologit) denom(cons) basecategory(4)) ///
> mcmc(orthogonal) initsprevious ///
> nopause nogroup

```

```

MLwiN 2.26 multilevel model                Number of obs      =      2867
Ordered multinomial logit response model
Estimation algorithm: MCMC

```

```

-----
      Contrast | Log-odds
-----+-----
           1 | 1 vs. 2 3 4
           2 | 1 2 vs. 3 4
           3 | 1 2 3 vs. 4
-----

Burnin                =          500
Chain                  =         5000
Thinning                =           1
Run time (seconds)    =          61.5
Deviance (dbar)       =        5874.57
Deviance (thetabar)   =        5826.78
Effective no. of pars (pd) =         47.79
Bayesian DIC           =        5922.36
-----

```

	Mean	Std. Dev.	ESS	P	[95% Cred. Interval]	
-----+-----						
Contrast 1						
cons_1	-3.536198	.1511914	77	0.000	-3.848515	-3.259399
-----+-----						
Contrast 2						
cons_2	-2.137839	.1398857	60	0.000	-2.430277	-1.880518
-----+-----						
Contrast 3						
cons_3	-1.645083	.1367751	68	0.000	-1.926747	-1.393776
-----+-----						
age_123	-.0150755	.0062137	1150	0.007	-.0275777	-.0031782
lc1_123	1.048174	.1251156	1140	0.000	.8017691	1.295756
lc2_123	1.382455	.1331677	997	0.000	1.124834	1.64314
lc3plus_123	1.281889	.1411943	1187	0.000	1.005635	1.563928
urban_123	.6650735	.0924569	612	0.000	.480046	.8432043
-----+-----						

Random-effects Parameters	Mean	Std. Dev.	ESS	[95% Cred. Int]	
Level 2: district					
var(cons_123)	.2615968	.0794832	426	.1442128	.4547639

Age has a significant negative effect and so younger women are significantly more likely to use more effective approaches to birth control than older women. All three living children dummy variables are positive and significant and so women with children, particularly those with two or three children, appear significantly more likely to use more effective approaches to birth control than childless women. Urban also has a significant positive effect and so women living in urban areas also appear more likely to use more effective means of birth control.

4.3. Nominal response model

In this example we fit a nominal logistic model to the Bangladeshi contraceptive use data. Specifically, we fit a two-level women-within-districts nominal response model to the four category contraceptive use variable `use4`. The model is a random intercept model where we adjust for the same variables as in our previous example and we again set the fourth category of `use4` (no contraception) to be the reference category. Thus, the model has three equations contrasting the log-odds of using each method of contraception against the reference of using no contraception.

$$\text{use4}_{ij} \sim \text{Multinomial}(\pi_{ij}^{(1)}, \pi_{ij}^{(2)}, \pi_{ij}^{(3)})$$

$$\log(\pi_{ij}^{(1)}/\pi_{ij}^{(4)}) = \beta_0^{(1)} + \beta_1^{(1)} \text{age}_{ij} + \beta_2^{(1)} \text{lc1}_{ij} + \beta_3^{(1)} \text{lc2}_{ij} + \beta_4^{(1)} \text{lc3plus}_{ij} + \beta_5^{(1)} \text{urban}_{ij} + u_j^{(1)}$$

$$\log(\pi_{ij}^{(2)}/\pi_{ij}^{(4)}) = \beta_0^{(2)} + \beta_1^{(2)} \text{age}_{ij} + \beta_2^{(2)} \text{lc1}_{ij} + \beta_3^{(2)} \text{lc2}_{ij} + \beta_4^{(2)} \text{lc3plus}_{ij} + \beta_5^{(2)} \text{urban}_{ij} + u_j^{(2)}$$

$$\log(\pi_{ij}^{(3)}/\pi_{ij}^{(4)}) = \beta_0^{(3)} + \beta_1^{(3)} \text{age}_{ij} + \beta_2^{(3)} \text{lc1}_{ij} + \beta_3^{(3)} \text{lc2}_{ij} + \beta_4^{(3)} \text{lc3plus}_{ij} + \beta_5^{(3)} \text{urban}_{ij} + u_j^{(3)}$$

$$\begin{pmatrix} u_j^{(1)} \\ u_j^{(2)} \\ u_j^{(3)} \end{pmatrix} \sim \text{N} \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u(1)}^2 & & \\ \sigma_{u(1,2)} & \sigma_{u(2)}^2 & \\ \sigma_{u(1,3)} & \sigma_{u(2,3)} & \sigma_{u(3)}^2 \end{pmatrix} \right\}$$

Fitting this model first by IGLS MQL1 and then by MCMC produces

```
. quietly runmlwin use4 cons age lc1 lc2 lc3plus urban, ///
> level2(district: cons, residuals(u)) ///
> level1(woman:) ///
> discrete(dist(multinomial) link(mlogit) denom(cons) basecategory(4)) ///
> nopause
.
. runmlwin use4 cons age lc1 lc2 lc3plus urban, ///
> level2(district: cons) ///
> level1(woman:) ///
> discrete(dist(multinomial) link(mlogit) denom(cons) basecategory(4)) ///
> mcmc(orthogonal) initsprevious ///
> nopause nogroup
```

```

MLwiN 2.26 multilevel model                    Number of obs      =      2867
Unordered multinomial logit response model
Estimation algorithm: MCMC

```

```

-----
Contrast | Log-odds
-----+-----
          1 | 1 vs. 4
          2 | 2 vs. 4
          3 | 3 vs. 4
-----

```

```

Burnin              =      500
Chain               =     5000
Thinning            =          1
Run time (seconds) =      170
Deviance (dbar)    =   5582.35
Deviance (thetabar) =   5479.14
Effective no. of pars (pd) = 103.21
Bayesian DIC       =   5685.55

```

```

-----
                |      Mean   Std. Dev.   ESS   P   [95% Cred. Interval]
-----+-----
Contrast 1     |
  cons_1      |  -4.082737  .3217545   222   0.000  -4.748469  -3.523268
  age_1       |   .0220265  .0096253  1326   0.012   .0030931   .0408361
  lc1_1       |   2.190912  .3368988   317   0.000   1.573424   2.887145
  lc2_1       |   2.647792  .3368055   303   0.000   2.003088   3.326137
  lc3plus_1   |   2.436404  .3372546   349   0.000   1.789113   3.12222
  urban_1     |   .2708482  .1652604   695   0.041  -.0449046   .603516
-----+-----
Contrast 2     |
  cons_2      |  -2.686565  .1658037   297   0.000   -3.0176   -2.369953
  age_2       |  -.0670038  .0093544   965   0.000  -.0850417  -.0481877
  lc1_2       |   1.070755  .1540116  1030   0.000   .7800555   1.379321
  lc2_2       |   1.375661  .1791258   995   0.000   1.022076   1.713762
  lc3plus_2   |   1.328719  .1933557   997   0.000   .9555698   1.712443
  urban_2     |   1.2434    .1220864   548   0.000   .9994968   1.48343
-----+-----
Contrast 3     |
  cons_3      |  -2.789423  .1983549   537   0.000   -3.186918  -2.409306
  age_3       |   .000286   .0099567  1140   0.485  -.0193697   .0192972
  lc1_3       |   .7973608  .234925    797   0.001   .3292199   1.257956
  lc2_3       |   1.15957   .2321483   940   0.000   .7110318   1.613051
  lc3plus_3   |   1.206419  .2372543   880   0.000   .7445371   1.672309
  urban_3     |   .2561048  .1722062   777   0.070  -.0845745   .5930236
-----

```

```

-----
Random-effects Parameters |      Mean   Std. Dev.   ESS   [95% Cred. Int]
-----

```

```

-----+-----
Level 2: district |
      var(cons_1) | .5185827 .1566131 364 .2756147 .8967777
    cov(cons_1,cons_2) | .2668872 .0970831 261 .1126262 .4930946
      var(cons_2) | .2783497 .0927847 189 .1362191 .4950176
    cov(cons_1,cons_3) | .2278537 .1053358 359 .0573296 .4664064
    cov(cons_2,cons_3) | .0964096 .0780254 325 -.0387304 .271509
      var(cons_3) | .3636968 .1254203 346 .171915 .6547303
-----+-----

```

To aid our interpretation of the model results, we can calculate and then graph predicted probabilities of using each method of contraception. The commands to generate these predictions and graphs are somewhat more complex than those presented for earlier examples and so we do not describe them here (see Supplementary materials for full details).

Figure 9 illustrates how the predicted probability of using each method of contraception changes by woman's age and whether that woman lives in a rural or urban area. (Predictions are calculated for a woman with two living children living in the average state). Figure 10 illustrates how the predicted probability of using each method of contraception changes by the number of living children a woman has, whether that woman lives in a rural or urban area and by which district the women lives in. (Predictions are calculated for a 30 year-old woman).

Figure 9 suggests that the probability of not using contraception increases with age. Among the three alternative methods of contraception, the probability of using traditional methods appears to decrease with age while the probability of using modern reversible method and particularly sterilization increases with age. The figure also suggests that the probability of using no contraception is much lower in urban areas than it is in rural areas while the probability of using traditional method is much higher in urban areas than it is in rural areas. The probabilities of using modern reversible methods and sterilization appear similar for women living in rural and urban areas.

Figure 10 suggests that the probability of not using contraception is highest for childless women, reduces for women with one child and is slightly lower again for women with two or more children. The probability of using each of the three different methods of contraception are all substantially higher for women with living children, particularly for those women with two or three living children, than they are for childless women. As in Figure 9, we see that women living in urban areas are much more likely to use contraception than women living in rural areas. Plotting the district specific probabilities of using each method of contraception, illustrates the substantial heterogeneity in woman's contraceptive use between districts; in some districts women are far more likely to use contraception than in other districts.

Finally, in the random part parameters table of the model output, the random effects correlations are all positive suggesting that districts with high (low) use of one type of method relative to using no contraception also tend to have high (low) use of other methods relative to using no contraception. The highest correlation is between the use of sterilization and the use of modern reversible methods which is expected as both of these types of method are promoted by family planning programs.

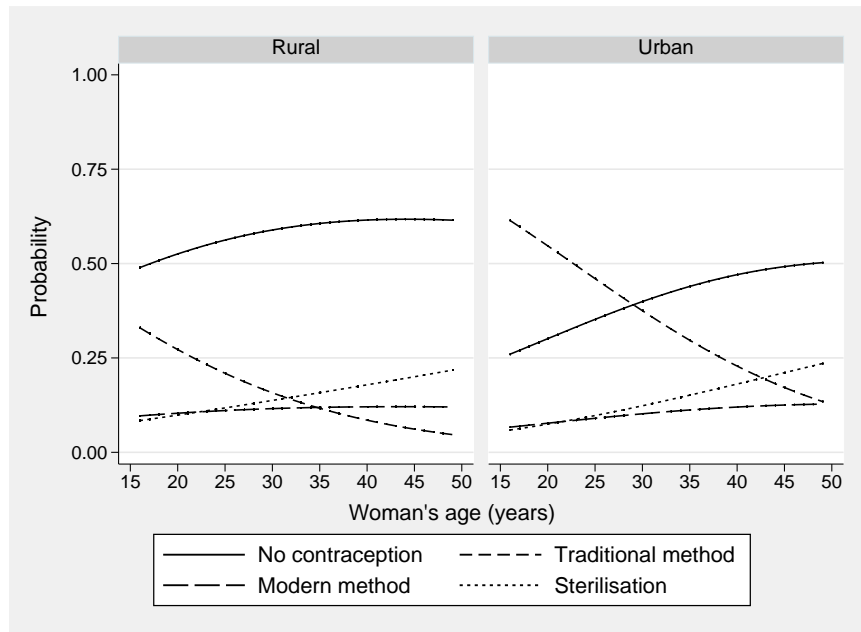


Figure 9: Category probabilities against age plotted in Stata.

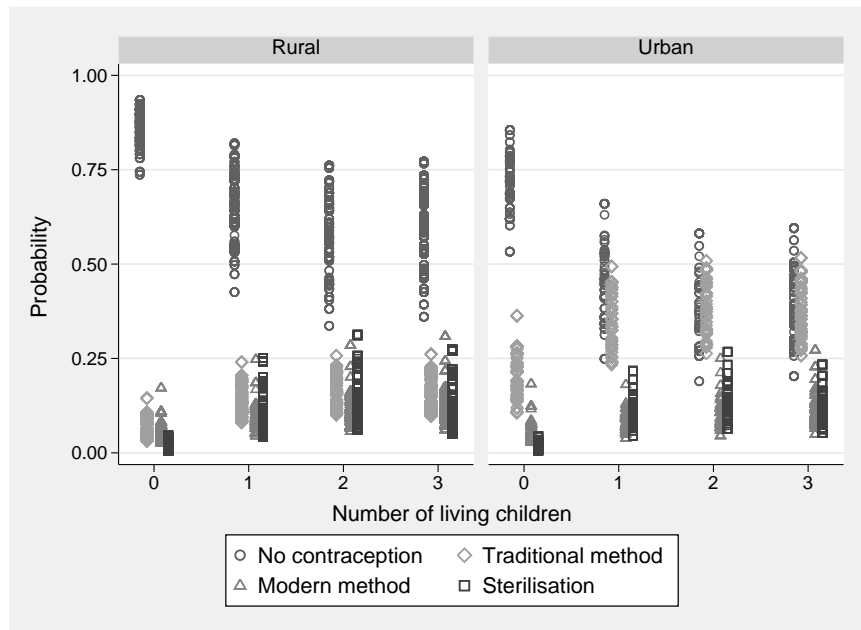


Figure 10: Category probabilities against number of living children plotted in Stata.

5. Conclusions

We have illustrated how to run **MLwiN** seamlessly from within **Stata** using the `runmlwin` command. The `runmlwin` command allows all models fitted in **MLwiN** to now be fitted within **Stata** and therefore greatly expands the range of multilevel modeling options available to **Stata** users. We note that where there is overlap with **Stata**'s existing commands, `runmlwin` will

often fit these models considerably more quickly. A very important advantage of operating **MLwiN** via `runmlwin` as opposed to through its point-and-click interface is that this makes carrying out reproducible and fully documented analyzes extremely easy. The annotated do-file which accompanies this article and replicates all of the presented **Stata** output is a case in point. More broadly, running **MLwiN** from within **Stata** allows users to take advantage of **Stata**'s excellent statistics, graphics and data management commands to prepare and descriptively analyze their multilevel datasets. After fitting a model by `runmlwin`, users can apply **Stata**'s comprehensive hypothesis testing, model comparison, and multilevel graphics facilities to aid model interpretation and to communicate their results.

We have created a dedicated `runmlwin` web site to provide users with further information and resources (<http://www.bristol.ac.uk/cmm/software/runmlwin/>). Readers can learn more about the wide range of models which can be fitted by downloading the different conference presentations we have given. Readers seeking more detail can then download **Stata** do-files and log-files which allow them to replicate every example presented in the two **MLwiN** manuals. A complete description of the command syntax and every modeling and estimation option is provided in the 28-page help file (Leckie and Charlton 2013). Finally, the website provides an active user forum where we encourage users to post their queries.

Acknowledgments

This research was funded under the LEMMA2 project, a node of the UK Economic and Social Research Council's National Centre for Research Methods (grant number RES-576-25-0003). We are very grateful to the helpful comments made by the two reviewers.

References

- Amin S, Diamond I, Steele F (1997). "Contraception and Religiosity in Bangladesh." In JC Caldwell, RMDGW Jones, RM D'Souza (eds.), *The Continuing Demographic Transition*, pp. 268–289. Oxford: Oxford University Press.
- Browne WJ (2012). *MCMC Estimation in MLwiN, v2.26*. Centre for Multilevel Modelling, University of Bristol, UK.
- Carpenter JR, Goldstein H, Kenward MG (2011). "REALCOM-IMPUTE Software for Multilevel Multiple Imputation with Mixed Response Types." *Journal of Statistical Software*, **45**(5), 1–14. URL <http://www.jstatsoft.org/v45/i05/>.
- Crouchley R, Stott D, Pritchard J (2009). *Multivariate Generalised Linear Mixed Models via sabreStata (Sabre in Stata) Version 1 (Draft)*. Centre for e-Science, Lancaster University.
- de Leeuw J, Meijer E (eds.) (2008). *Handbook of Multilevel Analysis*. Springer-Verlag.
- Goldstein H (1986). "Multilevel Mixed Linear Model Analysis Using Iterative Generalised Least Squares." *Biometrika*, **73**, 43–56.
- Goldstein H (2011). *Multilevel Statistical Models*. 4th edition. John Wiley & Sons, London.

- Goldstein H, Rasbash J, Yang M, Woodhouse G, Pan H, Nuttall D, Thomas S (1993). “A Multilevel Analysis of School Examination Results.” *Oxford Review of Education*, **19**, 425–433.
- Hox JJ (2010). *Multilevel Analysis: Techniques and Applications*. 2nd edition. Hogrefe and Huber.
- Hox JJ, Roberts JK (eds.) (2010). *Handbook of Advanced Multilevel Analysis*. Routledge, New York.
- Huq NM, Cleland J (1990). *Bangladesh Fertility Survey, 1989*. National Institute of Population Research and Training (NIPORT), Dhaka.
- Leckie G, Charlton C (2013). *runmlwin Help File*. University of Bristol, UK. URL <http://www.bristol.ac.uk/cmm/media/runmlwin/runmlwin.pdf>.
- Muthén LK, Muthén BO (2012). *Mplus User’s Guide*. Muthén & Muthén, Los Angeles, CA, 7th edition. URL <http://www.statmodel.com/>.
- Rabe-Hesketh S, Skrondal A (2012a). *Multilevel and Longitudinal Modeling Using Stata Volume I: Continuous Responses*. 3rd edition. Stata Press, College Station, TX.
- Rabe-Hesketh S, Skrondal A (2012b). *Multilevel and Longitudinal Modeling Using Stata Volume II: Categorical Responses, Counts, and Survival*. 3rd edition. Stata Press, College Station, TX.
- Rabe-Hesketh S, Skrondal A, Pickles A (2004). “**GLLAMM** Manual.” *Technical Report 160*, Working Paper Series, UC Berkeley Division of Biostatistics.
- Rasbash J, Charlton C, Browne WJ, Healy M, Cameron B (2009). *MLwiN Version 2.1*. Centre for Multilevel Modelling. URL <http://www.MLwiN.com/>.
- Rasbash J, Steele F, Browne WJ, Goldstein H (2012). *A User’s Guide to MLwiN, v2.26*. Centre for Multilevel Modelling, University of Bristol, UK.
- Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. 2nd edition. Sage, Thousand Oaks, CA.
- Raudenbush SW, Bryk AS, Congdon R (2012). *HLM 7 : Hierarchical Linear and Nonlinear Modeling*. Lincolnwood, IL. URL <http://www.hlm.com/>.
- Snijders TAB, Bosker RJ (2012). *Multilevel Analysis: an Introduction to Basic and Advanced Multilevel Modeling*. 2nd edition. Sage.
- Spiegelhalter DJ, Thomas A, Best NG, Lunn D (2003). *WinBUGS Version 1.4 User Manual*. MRC Biostatistics Unit, Cambridge. URL <http://www.mrc-bsu.cam.ac.uk/bugs/>.
- StataCorp (2011). *Stata Data Analysis Statistical Software: Release 12*. StataCorp LP, College Station, TX. URL <http://www.stata.com/>.
- Thompson J, Palmer T, Moreno S (2006). “Bayesian Analysis in Stata Using **WinBUGS**.” *The Stata Journal*, **6**, 530–549.

Affiliation:

George Leckie

Centre for Multilevel Modelling

Graduate School of Education

2 Priory Road

Bristol, BS8 1TX, United Kingdom

E-mail: g.leckie@bristol.ac.uk

URL: <http://www.bristol.ac.uk/cmm/team/leckie.html>