Running MLwiN from within Stata: the `runmlwin` command

CCSR/ISC Seminar series
Manchester
25th September 2012

George Leckie and Chris Charlton
Centre for Multilevel Modelling
University of Bristol
What is runmlwin?

• **runmlwin** is a Stata command to run MLwiN seamlessly from within Stata
  
  – MLwiN offers fast estimation of a wide range of multilevel models, but has limited data management, graphics and programming facilities
  
  – Stata offers a limited range of multilevel models, but has excellent facilities for pre- and post-estimation data management and graphics and many model testing and interpretation routines
  
  – **runmlwin** capitalises on the best features of both packages

• But what if you use R rather than Stata...
  
  – Then use the **R2MLwiN** R function to run MLwiN from within R (see later)
  
  – **R2MLwiN** provides all the same functionality as **runmlwin**
Multilevel modelling in Stata

• Stata provide the `xtmixed`, `xtmelogit` and `xtmepoisson` commands
  – Limited range of models can be specified
  – Computationally quite slow

• Sophia Rabe-Hesketh and colleagues have developed the `gllamm` command
  – Wide range of models can be specified
  – Computationally very slow

• Other user-written multilevel modelling commands available in Stata include: `hlm`, `realcomimpute`, `runmplus`, `sabrestata`, `winbugs`
Multilevel modelling in MLwiN

1. Estimation of multilevel models for continuous, binary, ordered categorical, unordered categorical and count data

2. Fast estimation via classical and Bayesian methods

3. Estimation of multilevel models for cross-classified and multiple membership non-hierarchical data structures

4. Estimation of multilevel multivariate response models, multilevel spatial models, multilevel measurement error models, multilevel multiple imputation models and multilevel factor models

5. Free to UK academics, thanks to ESRC funding
Outline

1. Continuous response models
2. Working efficiently
3. Binary response models
4. Simulation studies
5. MCMC estimation
6. Export models to WinBUGS
7. Speed comparisons
8. More complex analyses
9. Resources to help you learn runmlwin
10. Running MLwiN from within R: the R2MLwiN function
1. CONTINUOUS RESPONSE MODELS
Two-level variance components model

- Inner-London schools exam scores data set
- Main MLwiN User Manual example (the ‘tutorial’ data set)
- 4059 students nested within 65 schools

\[
\text{normexam}_{ij} = \beta_0 + u_j + e_{ij}
\]

\[
u_j \sim N(0, \sigma_u^2)
\]

\[
e_{ij} \sim N(0, \sigma_e^2)
\]
Statistics/Data Analysis

MP - Parallel Edition

12.1 Copyright 1985-2011 StataCorp LP
StataCorp
4905 Lakeway Drive
College Station, Texas 77845 USA
800-STATA-PC http://www.stata.com
979-696-4600 stata@stata.com
979-696-4601 (fax)

2-user 2-core Stata network perpetual license:
Serial number: 50120527735
Licensed to: ZoneA
University of Bristol

Notes:
1. (\v# option or -set maxvar-) 5000 maximum variables

running C:\Program Files (x86)\Stata12\sysprofile.do ...

running C:\Users\gl9158\profile.do ...

.
use http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta
Statistics/Data Analysis

MP - Parallel Edition

Copyright 1985-2011 StataCorp LP
StataCorp
4905 Lakeway Drive
College Station, Texas 77845 USA
800-STATAPC http://www.stata.com
979-696-4600 stata@stata.com
979-696-4601 (fax)

2-user 2-core Stata network perpetual license:
Serial number: 50120527735
Licensed to: ZoneA
University of Bristol

Notes:
1. (*/# option or -set maxvar-) 5000 maximum variables

running C:\Program Files (x86)\Stata12\sysprofile.do ...
running C:\Users\gl9158\profile.do ...

use http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta

.
The `runmlwin` command syntax

\[ \text{normexam}_{ij} = \beta_0 + u_j + e_{ij} \]

\[ u_j \sim N(0, \sigma_u^2) \]

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

. runmlwin normexam cons, ///

    level2(school: cons) ///

    level1(student: cons)
2-user 2-core Stata network perpetual license:
Serial number: 50120527735
Licensed to: ZoneA
University of Bristol

Notes:
1. (/v# option or -set maxvar-) 5000 maximum variables

running C:\Program Files (x86)\Stata12\sysprofile.do ...
running C:\Users\g19158\profile.do ...
.use http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta

runmlwin normexam cons, level2(school: cons) level1(student: cons)
MLwiN
Version 2.25
© Centre for Multilevel Modelling
University of Bristol
Software authors:
Jon Rasbash
and
William Browne
Michael Healy
Bruce Cameron
Christopher Charlton
February 2012
We are grateful to the ESRC for their sustained support.
\[ \text{normexam}_{ij} \sim N(XB, \Omega) \]
\[ \text{normexam}_{ij} = \beta_{0ij}\text{cons} \]
\[ \beta_{0ij} = \beta_0 + u_{0j} + e_{0ij} \]

\[
\begin{bmatrix}
u_{0j} \\
u_{0j}
\end{bmatrix} \sim N(0, \Omega_u) : \Omega_u = \begin{bmatrix} \sigma^2_u & 0 \\ 0 & \sigma^2_u \end{bmatrix}
\]

\[
\begin{bmatrix}e_{0ij} \\
e_{0ij}
\end{bmatrix} \sim N(0, \Omega_e) : \Omega_e = \begin{bmatrix} \sigma^2_e & 0 \\ 0 & \sigma^2_e \end{bmatrix}
\]

(4059 of 4059 cases in use)
normexam\(_{ij}\) \sim N(XB, \ \Omega)

normexam\(_{ij}\) = \beta_{0ij}\text{cons}

\beta_{0ij} = -0.013(0.054) + u_{0j} + e_{0ij}

\begin{bmatrix} u_{0j} \end{bmatrix} \sim N(0, \ \Omega_u) : \ \Omega_u = \begin{bmatrix} 0.169(0.032) \end{bmatrix}

\begin{bmatrix} e_{0ij} \end{bmatrix} \sim N(0, \ \Omega_e) : \ \Omega_e = \begin{bmatrix} 0.848(0.019) \end{bmatrix}

-2*\loglikelihood(IGLS Deviance) = 11010.648(4059 of 4059 cases in use)
. use http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta

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

MLwiN 2.25 multilevel model
Number of obs = 4059
Normal response model
Estimation algorithm: IGLS

<table>
<thead>
<tr>
<th>Level Variable</th>
<th>No. of Groups</th>
<th>Observations per Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>school</td>
<td>65</td>
<td>2 62.4 198</td>
</tr>
</tbody>
</table>

Run time (seconds) = 17.13
Number of iterations = 3
Log likelihood = -5505.3242
Deviance = 11010.648

| normexam       | Coef.    | Std. Err. | z     | P>|z|  | [95% Conf. Interval] |
|----------------|----------|-----------|-------|------|---------------------|
| cons           | -0.0131668 | 0.0536254 | -0.25 | 0.806 | -0.1182706 0.091937 |

<table>
<thead>
<tr>
<th>Random-effects Parameters</th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2: school</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>0.1686251</td>
<td>0.0324466</td>
<td>0.1050309 0.2322194</td>
</tr>
<tr>
<td>Level 1: student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>0.8477613</td>
<td>0.0189712</td>
<td>0.8105786 0.8849441</td>
</tr>
</tbody>
</table>
. runlwin normexam cons, level2(school: cons) level1(student: cons)

MLwiN 2.25 multilevel model
Normal response model
Estimation algorithm: IGLS

<table>
<thead>
<tr>
<th>Level Variable</th>
<th>No. of Groups</th>
<th>Observations per Group</th>
<th>Minimum</th>
<th>Average</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>school</td>
<td>65</td>
<td>2</td>
<td>62.4</td>
<td>198</td>
<td></td>
</tr>
</tbody>
</table>

Run time (seconds) = 17.13
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 | .806 | -.1182706 .091937    |

Random-effects Parameters

<table>
<thead>
<tr>
<th>Level 2: school</th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>var(cons)</td>
<td>.1686251</td>
<td>.0324466</td>
<td>.1050309 .2322194</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level 1: student</th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>var(cons)</td>
<td>.8477613</td>
<td>.0189712</td>
<td>.8105786 .8849441</td>
</tr>
</tbody>
</table>
Add covariates

\[ \text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_j + e_{ij} \]

\[ u_j \sim \text{N}(0, \sigma_u^2) \]

\[ e_{ij} \sim \text{N}(0, \sigma_e^2) \]

. runmlwin normexam cons standlrt girl, ///

   level2(school: cons) ///

   level1(student: cons)
Include a random slope

\[ \text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} + e_{ij} \]

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

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

. runmlwin normexam cons standlrt girl, ///
  level2(school: cons standlrt) ///
  level1(student: cons)
Allow for level 1 heteroskedasticity

\[ \text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} \]

\[ + e_{2ij} \text{girl}_{ij} + e_{3ij} \text{boy}_{ij} \]

\[ (u_{0j}) \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u0}^2 & \sigma_{u01}^2 \\ \sigma_{u01}^2 & \sigma_{u1}^2 \end{pmatrix} \right) \]

\[ (e_{2ij}, e_{3ij}) \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{e2}^2 & 0 \\ 0 & \sigma_{e3}^2 \end{pmatrix} \right) \]

. generate boy = 1 - girl

. runmlwin normexam cons standlrt girl, ///

   level2(school: cons standlrt) ///

   level1(student: girl boy, diagonal)
Retrieve the level 2 residuals

\[ \text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} \]

\[ + e_{2ij} \text{girl}_{ij} + e_{3ij} \text{boy}_{ij} \]

\[ (u_{0j}) \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{u0} & \sigma^2_{u1} \\ \sigma^2_{u01} & \sigma^2_{u1} \end{pmatrix} \right) \]

\[ (e_{2ij}) \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{e2} & 0 \\ 0 & \sigma^2_{e3} \end{pmatrix} \right) \]

. runmlwin normexam cons standlrt girl, ///

level2(school: cons standlrt, residuals(u)) ///

level1(student: girl boy, diagonal)
Do not pause in MLwiN and do not display the group table in Stata

\[
\begin{align*}
normexam_{ij} &= \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} \\
&\quad + e_{2ij} \text{girl}_{ij} + e_{3ij} \text{boy}_{ij} \\
\end{align*}
\]

\[
\begin{align*}
(u_{0j}) &\sim N\left(0, \begin{pmatrix} \sigma_{u0}^2 \\ \sigma_{u01} \end{pmatrix}\right) \\
(u_{1j}) &\sim N\left(0, \begin{pmatrix} \sigma_{u1}^2 \\ \sigma_{u01} \end{pmatrix}\right) \\
(e_{2ij}) &\sim N\left(0, \begin{pmatrix} \sigma_{e2}^2 \\ 0 \end{pmatrix}\right) \\
(e_{3ij}) &\sim N\left(0, \begin{pmatrix} \sigma_{e3}^2 \\ 0 \end{pmatrix}\right)
\end{align*}
\]

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

MLwiN 2.25 multilevel model
Number of obs = 4059

Normal response model
Estimation algorithm: IGLS
Run time (seconds) = 1.84
Number of iterations = 4
Log likelihood = -4640.71
Deviance = 9281.4199

|      | Coef.  | Std. Err. |    z  | P>|z| | [95% Conf. Interval] |
|------|--------|-----------|-------|------|----------------------|
| normexam |       |           |       |      |                      |
| cons   | -.111534 | .0433072  | -2.58 | .010 | -.1964145 – .0266536 |
| standlrt | .5529361 | .0200758  | 27.54 | .000 | .5135882 – .5922841  |
| girl   | .1752785 | .0324156  | 5.41  | .000 | .1117451 – .238812   |

Random-effects Parameters
<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2: school</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>.0862511</td>
<td>.017175</td>
<td>.0525887 – .1199135</td>
</tr>
<tr>
<td>cov(cons,standlrt)</td>
<td>.0190537</td>
<td>.0066789</td>
<td>.0059632 – .0321441</td>
</tr>
<tr>
<td>var(standlrt)</td>
<td>.0148919</td>
<td>.0044702</td>
<td>.0061304 – .0236534</td>
</tr>
<tr>
<td>Level 1: student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(girl)</td>
<td>.5251641</td>
<td>.0152836</td>
<td>.4952088 – .5551194</td>
</tr>
<tr>
<td>var(boy)</td>
<td>.5874345</td>
<td>.0209983</td>
<td>.5462786 – .6285904</td>
</tr>
</tbody>
</table>
. runmlwin normexam cons standlrt girl, ///
>   level2(school: cons standlrt, residuals(u)) ///
>   level1(student: girl boy, diagonal) ///
>   nogroup nopause

MLwiN 2.25 multilevel model
Normal response model
Estimation algorithm: IGLS
Run time (seconds) = 1.84
Number of iterations = 4
Log likelihood = -4640.71
Deviance = 9281.4199

|               | Coef.  | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|---------------|--------|-----------|-------|-----|----------------------|
| normexam      |        |           |       |     |                      |
| cons          | -.111534 | .0433072 | -2.58 | 0.010 | -.1964145 -.0266536 |
| standlrt      | .5529361 | .0200758 | 27.54 | 0.000 | .5135882 .5922841   |
| girl          | .1752785 | .0324156 | 5.41  | 0.000 | .1117451 .238812    |

Random-effects Parameters

<table>
<thead>
<tr>
<th></th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2: school</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>.0862511</td>
<td>.017175</td>
<td>.0525887 .1199135</td>
</tr>
<tr>
<td>cov(cons,standlrt)</td>
<td>.0190537</td>
<td>.0066789</td>
<td>.0059632 .0321441</td>
</tr>
<tr>
<td>var(standlrt)</td>
<td>.0148919</td>
<td>.0044702</td>
<td>.0061304 .0236534</td>
</tr>
<tr>
<td>Level 1: student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(girl)</td>
<td>.5251641</td>
<td>.0152836</td>
<td>.4952088 .5551194</td>
</tr>
<tr>
<td>var(boy)</td>
<td>.5874345</td>
<td>.0209983</td>
<td>.5462786 .6285904</td>
</tr>
</tbody>
</table>
. bysort school: keep if _n==1
. egen u0rank = rank(u0)
. serrbar u0 u0se u0rank, scale(1.96) yline(0)
. gen yhat = [FP1]cons + [FP1]stand*stand + u0 + ul*stand
. sort school standlrt
. line yhat standlrt, connect(ascending)
. lrtest model1 model2

Likelihood-ratio test
(Assumption: model1 nested in model2)

LR chi2(5) =  1729.23
Prob > chi2 =  0.0000

.
.
.

.test [RP1]var(girl) = [RP1]var(boy)

( 1)  [RP1]var(girl) - [RP1]var(boy) = 0

    chi2(  1) =  5.74
    Prob > chi2 =  0.0166

.
.
.

.nlcom (Boy_VPC_xis0: [RP2]var(cons)/([RP2]var(cons) + [RP1]var(boy)))

Boy_VPC_xis0:  [RP2]var(cons)/([RP2]var(cons) + [RP1]var(boy))

|            | Coef.     | Std. Err. |    z |    P>|z| | [95% Conf. Interval] |
|------------|-----------|-----------|------|----------|----------------------|
|Boy_VPC_xis0| .1280287  | .0226244  | 5.66 | 0.000    | .0836856 .1723718    |
2. STATA MAKES IT EASY TO WORK EFFICIENTLY
* 1. TWO-LEVEL MULTILEVEL MODELS

* Open the tutorial data set
use "http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta", clear

* Fit a two-level (students within schools) variance components model to
* a continuous educational response variable, normexam. Note, you will need
* to click the "Resume Macro" button twice in MLwiN to return the model
* results to the Stata output window.
runmlwin normexam cons, ///
   level2(school: cons) ///
   level1(student: cons)

* Store the model estimates
estimates store model1

* Generate a boy dummy variable
generate boy = 1 - girl

* Extend the previous model to include fixed part covariates, a random school
* level slope and separate level 1 residuals for boys and girls. The runmlwin
* command also requests that runmlwin extracts the predicted values for the
* school level residuals from MLwiN and returns them to Stata. The nopause
* option prevents MLwiN from pausing before and after model estimation and so
* returns the model results automatically to Stata.
runmlwin normexam cons standlrt girl, ///
   level2(school: cons standlrt, residuals(u)) ///
   level1(student: girl boy, diagonal) nopause

* Store the model estimates
estimates store model2

* Perform a likelihood ratio test to compare the boy and girl residual
* variances
3. BINARY RESPONSE MODELS
Random slope logistic model

\[ \text{passexam}_{ij} \sim \text{Binomial}(1, \pi_{ij}) \]

\[ \logit(\pi_{ij}) = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_0 + u_1 \text{standlrt}_{ij} \]

\[ \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{u0} & \sigma_{u01} \\ \sigma_{u01} & \sigma^2_{u1} \end{pmatrix} \right) \]

. generate passexam = (normexam>0)

. runmlwin passexam cons standlrt girl, ///
    level2(school: cons standlrt) ///
    level1(student:) ///
    discrete(dist(binomial) link(logit) denom(cons)) ///
    nogroup nopause
. generate passexam = (normexam>0)

. runmlwin passexam cons standlrt girl, ///
  level2(school: cons standlrt) ///
  level1(student:) ///
  discrete(distribution(binomial) link(logit) denominator(cons)) ///
  nogroup nopause

MLwiN 2.25 multilevel model
Binomial logit response model
Estimation algorithm: IGLS, MQLI
Run time (seconds) = 1.61
Number of iterations = 6

+-----------------------------------+-----------------+-----------------+-----------------+-----------------+-----------------+
|                  | Coef. | Std. Err. | z    | P>|z| | [95% Conf. Interval] |
|-------------------|-------|-----------|------|------|----------------------|
| passexam          |       |           |      |      |                      |
| cons              | -0.0479964 | 0.101761 | -0.47 | 0.637 | -0.2474444 | 0.1514515 |
| standlrt          | 1.232918  | 0.0581067 | 21.22 | 0.000 | 1.1190311 | 1.3468054 |
| girl              | 0.186636  | 0.0956229 | 1.95  | 0.051 | -0.0007814 | 0.3740534 |

+-----------------------------------+-----------------+-----------------+-----------------+-----------------+
<table>
<thead>
<tr>
<th>Random-effects Parameters</th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2: school</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>0.3701358</td>
<td>0.0822183</td>
<td>0.208991</td>
</tr>
<tr>
<td>cov(cons,standlrt)</td>
<td>0.0444551</td>
<td>0.0394446</td>
<td>-0.0328549</td>
</tr>
<tr>
<td>var(standlrt)</td>
<td>0.06152</td>
<td>0.0364277</td>
<td>-0.009877</td>
</tr>
</tbody>
</table>
Fit model by PQL2 using MQL1 estimates as starting values

\[ \text{passexam}_{ij} \sim \text{Binomial}(1, \pi_{ij}) \]

\[ \text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_0 + u_1 \text{standlrt}_{ij} \]

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

. runmlwin passexam cons standlrt girl, ///

level2(school: cons standlrt) ///

level1(student:) ///

discrete(d(binomial) l(logit) de(cons) pql2) ///

initsprevious nopause
. runmlwin passexam cons standlrt girl, ///
>   level2(school: cons standlrt) ///
>   level1(student:) ///
>   discrete(dist(binomial) link(logit) denom(cons) pql2) ///
>   initsprevious nogroup nopause

Model fitted using initial values specified as parameter estimates from previous model

MLwiN 2.25 multilevel model
Binomial logit response model
Estimation algorithm: **IGLS, PQL2**
Run time (seconds) = 2.04
Number of iterations = 8

| passexam  | Coef.   | Std. Err. | z      | P>|z| | [95% Conf. Interval] |
|------------|---------|-----------|--------|-----|---------------------|
| cons       | -.0367105 | .1120693  | -0.33  | 0.743 | -.2563622 to .1829413 |
| standlrt   | 1.358886  | .0642726  | 21.14  | 0.000 | 1.232914 to 1.484858 |
| girl       | .2012481  | .1013948  | 1.98   | 0.047 | .0025179 to .3999782 |

Random-effects Parameters

<table>
<thead>
<tr>
<th>Level 2: school</th>
<th>Estimate</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>var(cons)</td>
<td>.4740776</td>
<td>.1031501</td>
<td>.2719071 to .676248</td>
</tr>
<tr>
<td>cov(cons,standlrt)</td>
<td>.0625434</td>
<td>.0491646</td>
<td>-.0338175 to .1589043</td>
</tr>
<tr>
<td>var(standlrt)</td>
<td>.0764959</td>
<td>.0443148</td>
<td>-.0103596 to .1633514</td>
</tr>
</tbody>
</table>
```
estimates table mql1 pql2, ///
  stats(ll N) b(%4.3f) stfmt(%4.0f) varwidth(18) newpanel

<table>
<thead>
<tr>
<th>Variable</th>
<th>mql1</th>
<th>pql2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cons</td>
<td>-0.048</td>
<td>-0.037</td>
</tr>
<tr>
<td>standlrt</td>
<td>1.233</td>
<td>1.359</td>
</tr>
<tr>
<td>girl</td>
<td>0.187</td>
<td>0.201</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>0.370</td>
<td>0.474</td>
</tr>
<tr>
<td>cov(cons \ standlrt)</td>
<td>0.044</td>
<td>0.063</td>
</tr>
<tr>
<td>var(standlrt)</td>
<td>0.062</td>
<td>0.076</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(bcons_1)</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ll</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>4059</td>
<td>4059</td>
</tr>
</tbody>
</table>
```
4. SIMULATION STUDIES ARE NOW EASY
* REPLICATE RODRIGUEZ AND GOLDMAN (1995)

```stata
clear
set seed 12345
postutil clear
postfile MQL1 ix fx cx sigmaf sigmac using "MQL1.dta", replace
set obs 2
generate cx = _n - 1
expand 10
sort cx
generate cid = _n
expand 2
bysort cid: gen fx = _n - 1
expand 10
bysort cid (fx): generate fid = _n
expand 2
bysort cid fid: gen ix = _n - 1
expand 10
bysort cid fid (ix): gen iid = _n
generate cons = 1
forvalues iteration = 1/100 {
    display _n(5) as txt "Iteration " as res `iteration' as txt " of " as res "100"
generate c = rnormal(0,1)
bysort cid (fid iid): replace c = c[1]
generate f = rnormal(0,1)
bysort cid fid (iid): replace f = f[1]
generate y = rbinomial(1, invlogit(0*cons + 1*ix + 1*fx + 1*cx + f + c))
rumllwin y cons ix fx cx, level3(cid: cons) level2(fid: cons) level1(iid:)
            discrete(distribution(binomial) link(logit) denominator(cons))
            nopause
    post MQL1 ([FP1]ix) ([FP1]fx) ([FP1]cx) (sqrt([RP2]var(cons))) (sqrt([RP3]var(cons)))
drop c f y
}
postclose MQL1
use "MQL1.dta", clear
tabstat ix fx cx sigmaf sigmac, format(%3.2f)
```

5. MCMC ESTIMATION
Random slope logistic model

\[ \text{passexam}_{ij} \sim \text{Binomial}(1, \pi_{ij}) \]

\[ \logit(\pi_{ij}) = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} \]

\[ (u_{0j}, u_{1j}) \sim \text{N}\{ (0, 0), \begin{pmatrix} \sigma_{u0}^2 & 0 \\ 0 & \sigma_{u1}^2 \end{pmatrix} \} \]

. runmlwin passexam cons standlrt girl, ///
  level2(school: cons standlrt) ///
  level1(student:) ///
  discrete(d(binomial) l(logit) de(cons)) ///
  mcmc(burnin(500) chain(5000)) ///
  initsprevious nogroup nopause
. runmlwin passexam cons stand1lrt girl, ///
>   level2(school: cons stand1lrt) level1(student:): ///
>   discrete(d(binomial) l(logit) de(cons)): ///
>   mcmc(burnin(500) chain(5000)) inits(previous) nogroup nopause

MLwiN 2.25 multilevel model
Binomial logit response model
Estimation algorithm: MCMC

Burnin = 500
Chain = 5000
Thinning = 1
Run time (seconds) = 30.1
Deviance (dbar) = 4232.10
Deviance (thetabar) = 4159.24
Effective no. of pars (pd) = 72.86
Bayesian DIC = 4304.96

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>ESS</th>
<th>P</th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cons</td>
<td>-.0347943</td>
<td>.1073479</td>
<td>94</td>
<td>.381</td>
<td>-.2506524 .1779318</td>
</tr>
<tr>
<td>stand1lrt</td>
<td>1.35652</td>
<td>.0624149</td>
<td>496</td>
<td>.000</td>
<td>1.231931 1.480608</td>
</tr>
<tr>
<td>girl</td>
<td>.1873172</td>
<td>.1005095</td>
<td>196</td>
<td>.026</td>
<td>-.0023705 .3851183</td>
</tr>
</tbody>
</table>

Random-effects Parameters

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>ESS</th>
<th>[95% Cred. Int]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2: school</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>var(cons)</td>
<td>.5135376</td>
<td>.1199011</td>
<td>1030</td>
<td>.3204156 .7835187</td>
</tr>
<tr>
<td>cov(cons,stand1lrt)</td>
<td>.0668458</td>
<td>.0581714</td>
<td>198</td>
<td>-.0387322 .1982548</td>
</tr>
<tr>
<td>var(stand1lrt)</td>
<td>.0862781</td>
<td>.0467082</td>
<td>99</td>
<td>.0243268 .2023509</td>
</tr>
</tbody>
</table>
. mcmcs, trajectories
. mcmcsrum, densities
. mcmcsunm [RP2]var(standlrlt), fiveplot
. mcmcsum [RP2]var(standlrt), detail

[RP2]var(standlrt)

Percentiles

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
<th>Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.0862781</td>
<td>0.5%</td>
</tr>
<tr>
<td>MCSE of Mean</td>
<td>0.0024099</td>
<td>2.5%</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>0.0467082</td>
<td>5%</td>
</tr>
<tr>
<td>Mode</td>
<td>0.0631075</td>
<td>25%</td>
</tr>
<tr>
<td>P(mean)</td>
<td>0</td>
<td>50%</td>
</tr>
<tr>
<td>P(mode)</td>
<td>0</td>
<td>75%</td>
</tr>
<tr>
<td>P(median)</td>
<td>0</td>
<td>95%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>99.5%</td>
</tr>
</tbody>
</table>

- Thinned Chain Length: 5000
- Effective Sample Size: 99
- Raftery Lewis (2.5%): 25770
- Raftery Lewis (97.5%): 23976
- Brooks Draper (mean): 446390
. `runmlwin`, nogroup mode or sd correlation

MLwiN 2.25 multilevel model                                            Number of obs  =  4059
Binomial logit response model                                          
Estimation algorithm: MCMC
Burnin                      =   500
Chain                       =  5000
Thinning                    =    1
Run time (seconds)          =   30.2
Deviance (dbar)             =  4232.10
Deviance (thetabar)         =  4159.24
Effective no. of pars (pd)  =   72.86
Bayesian DIC                =  4304.96

<table>
<thead>
<tr>
<th></th>
<th>Odds Ratio</th>
<th>Std. Dev.</th>
<th>ESS</th>
<th></th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cons</td>
<td>.9579759</td>
<td>.1042228</td>
<td>93</td>
<td>.381</td>
<td>.7782928</td>
</tr>
<tr>
<td>standl1rt</td>
<td>3.884481</td>
<td>.2435176</td>
<td>497</td>
<td>.000</td>
<td>3.427844</td>
</tr>
<tr>
<td>girl</td>
<td>1.205355</td>
<td>.122038</td>
<td>195</td>
<td>.026</td>
<td>.9976324</td>
</tr>
</tbody>
</table>

Random-effects Parameters

<table>
<thead>
<tr>
<th>Mode</th>
<th>Std. Dev.</th>
<th>ESS</th>
<th>[95% Cred. Int]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 2: school</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sd(cons)</td>
<td>.7030943</td>
<td>.0822202</td>
<td>1013</td>
</tr>
<tr>
<td>corr(cons,standl1rt)</td>
<td>.3872765</td>
<td>.2455501</td>
<td>210</td>
</tr>
<tr>
<td>sd(standl1rt)</td>
<td>.2642264</td>
<td>.0760582</td>
<td>91</td>
</tr>
</tbody>
</table>
6. EXPORT MODELS TO WinBUGS
Random slope logistic model

\[ \text{passexam}_{ij} \sim \text{Binomial}(1, \pi_{ij}) \]

\[ \logit(\pi_{ij}) = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_0 + u_1 \text{standlrt}_{ij} \]

\[ \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u0}^2 & 0 \\ 0 & \sigma_{u1}^2 \end{pmatrix} \right) \]

. runmlwin passexam cons standlrt girl, ///
  level2(school: cons standlrt) ///
  level1(student:) ///
  discrete(d(binomial) l(logit) de(cons)) ///
  mcmc(b(500) c(5000) savewinbugs(model(m.txt) inits(i.txt) data(d.txt) nofit)) ///
  initsprevious nogroup nopause
WINBUGS 1.4 code generated from MLwiN program

MODEL Definition

model
{
# Level 1 definition
for(i in 1:N) {
passexam[i] ~ dbin(p[i],denom[i])
logit(p[i]) <- beta[1] * cons[i]
+ beta[2] * standlrt[i]
+ beta[3] * girl[i]
+ u2[school[i],1] * cons[i]
+ u2[school[i],2] * standlrt[i]
}
# Higher level definitions
for (j in 1:n2) {
u2[j,1:2] ~ dmnorm(zero2[1:2],tau.u2[1:2,1:2])
}
# Priors for fixed effects
for (k in 1:3) { beta[k] ~ dflat() }
# Priors for random terms
for (i in 1:2) { zero2[i] <- 0 }
tau.u2[1:2,1:2] ~ dwish(R2[1:2, 1:2],2)
sigma2.u2[1:2,1:2] <- inverse(tau.u2[1:2,1:2])
}
7. SPEED COMPARISONS
runmlwin vs. xtmixed

- Simulated data: 130,000 students in 650 schools (200 students per school)

\[
\text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} + e_{2ij} \text{girl}_{ij} + e_{3ij} \text{boy}_{ij}
\]

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

\[
\begin{pmatrix}
    e_{2ij} \\
    e_{3ij}
\end{pmatrix} \sim N\begin{pmatrix}
    0 \\
    0
\end{pmatrix}, \begin{pmatrix}
    \sigma_{e2}^2 & \sigma_{e23} \\
    \sigma_{e23} & \sigma_{e3}^2
\end{pmatrix}
\]

<table>
<thead>
<tr>
<th>Software method</th>
<th>Seconds</th>
<th>$\beta_0$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\sigma_{u0}^2$</th>
<th>$\sigma_{u01}$</th>
<th>$\sigma_{u1}^2$</th>
<th>$\sigma_{e2}^2$</th>
<th>$\sigma_{e3}^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True values</td>
<td>—</td>
<td>0.00</td>
<td>0.50</td>
<td>0.20</td>
<td>0.10</td>
<td>0.00</td>
<td>0.05</td>
<td>0.50</td>
<td>0.60</td>
</tr>
<tr>
<td>runmlwin</td>
<td>6</td>
<td>-0.01</td>
<td>0.50</td>
<td>0.20</td>
<td>0.10</td>
<td>0.01</td>
<td>0.05</td>
<td>0.50</td>
<td>0.60</td>
</tr>
<tr>
<td>xtmixed</td>
<td>158</td>
<td>-0.01</td>
<td>0.50</td>
<td>0.20</td>
<td>0.10</td>
<td>0.01</td>
<td>0.05</td>
<td>0.50</td>
<td>0.60</td>
</tr>
</tbody>
</table>
runmlwin vs. xtmelogit

\( \text{passexam}_{ij} \sim \text{Binomial}(1, \pi_{ij}) \)

\[
\text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_0 + u_1 \text{standlrt}_{ij}
\]

\[
\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{u0}^2 & \sigma_{u01}^2 \\ \sigma_{u01}^2 & \sigma_{u1}^2 \end{pmatrix} \right)
\]

<table>
<thead>
<tr>
<th>Software method</th>
<th>Seconds</th>
<th>( \beta_0 )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
<th>( \sigma_{u0}^2 )</th>
<th>( \sigma_{u01} )</th>
<th>( \sigma_{u1}^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True values</td>
<td>–</td>
<td>0.00</td>
<td>1.50</td>
<td>0.20</td>
<td>0.50</td>
<td>0.00</td>
<td>0.10</td>
</tr>
<tr>
<td>runmlwin, mql1</td>
<td>9</td>
<td>-0.01</td>
<td>1.32</td>
<td>0.18</td>
<td>0.40</td>
<td>0.00</td>
<td>0.13</td>
</tr>
<tr>
<td>runmlwin, pql2</td>
<td>14</td>
<td>-0.01</td>
<td>1.49</td>
<td>0.20</td>
<td>0.50</td>
<td>-0.00</td>
<td>0.12</td>
</tr>
<tr>
<td>runmlwin, b(200) c(1000)</td>
<td>313</td>
<td>0.00</td>
<td>1.49</td>
<td>0.21</td>
<td>0.50</td>
<td>-0.00</td>
<td>0.11</td>
</tr>
<tr>
<td>runmlwin, b(500) c(5000)</td>
<td>310</td>
<td>0.00</td>
<td>1.49</td>
<td>0.21</td>
<td>0.50</td>
<td>-0.00</td>
<td>0.11</td>
</tr>
<tr>
<td>xtmelogit, intpoints(1)</td>
<td>265</td>
<td>-0.01</td>
<td>1.49</td>
<td>0.20</td>
<td>0.50</td>
<td>-0.00</td>
<td>0.11</td>
</tr>
<tr>
<td>xtmelogit, intpoints(7)</td>
<td>451</td>
<td>-0.01</td>
<td>1.49</td>
<td>0.20</td>
<td>0.50</td>
<td>-0.00</td>
<td>0.12</td>
</tr>
</tbody>
</table>
8. MORE COMPLEX ANALYSES
Five interesting extensions

1. Use runmlwin to quickly obtain approximate quasilikelihood estimates for discrete response models; then finish off estimation using adaptive quadrature in gllamm

2. Use runmlwin to fit ‘disease mapping’ spatial multilevel models and then plot thematic maps of the area-level residuals using the spmap command

3. After fitting model by MCMC using runmlwin, use mcmcsun to pull back MCMC chains in order to derive posterior distribution for any function of the parameters and data of interest (e.g. ICC or ranks of random effects)

4. Use the realcomimpute command to generate multiply imputed data sets; then use the runmlwin command with the mi estimate prefix to fit the model of interest to each data set and to combine results using ‘Rubin’s rules’

5. Use runmlwin to generate WinBUGS model, data and initial values files for any MLwiN MCMC model; then fit the model in WinBUGS using the winbugs command; then interpret chains using the mcmcsun command
9. RESOURCES TO HELP YOU LEARN runmlwin
help runmlwin

Title

runmlwin - Run the MLwiN multilevel modelling software from within Stata

Syntax

runmlwin responses_and_fixed_part, random_part [discrete(discrete_options)] [mcmc(mcmc_options)]
  [general_options]

where the syntax of responses_and_fixed_part is one of the following

for univariate continuous, binary, proportion and count response models

  depvar indepvars [if] [in]

for univariate ordered and unordered categorical response models

  depvar indepvars1 [(indepvars2, contrast(numlist)) ...] [if] [in]

where indepvars1 are those independent variables which appear with separate coefficients in each of
every log-odds contrast, while indepvars2 are those independent variables which appear with common
coefficients for those log-odds contrasts specified in contrast(numlist). Contrasts can be thought of as the separate "subequations" or "arms" of a multinomial response model. These contrasts are
indexed 1,2,... up to the total number of contrasts included in the model. The total number of
contrasts will be one less than the number of response categories.

for multivariate response models
Estimation

(a) random part estimation options

All options reported in this sub-section are specific to the level

reset(resetname) specifies the action to be taken when during estimation is estimated to be negative. all resets associated covariances. variances resets a negative variance to zero and all covariances. none ignores negative variances; no parameter is changed.

(b) discrete response estimation options

mgll1, the default, specifies that the model be fitted using a first order linearization. See Remarks on quasi-likelihood estimates: Model

mgll2 specifies that the model be fitted using a second order

pgll1 specifies that the model be fitted using a first order penalized

pgll2 specifies that the model be fitted using a second order penalized

(c) MCMC estimation options

on fits the specified model using default MCMC options.

burnin(#) specifies the number of iterations for the burn-in period; the option specifies the number of iterations necessary for the parameters, equivalently, to converge to a stationary distribution.
Examples

IMPORTANT. The following examples will only work on your computer once you have installed MLwiN and once you have told runmlwin what the mlwin.exe file address is. See Remarks on runmlwin installation instructions above for more information.

(a) Continuous response models

Two-level models

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

Two-level random-intercept model, analogous to xtreg (fitted using IGLS)
(See page 28 of the MLwiN User Manual)
```
   . runmlwin normexam cons standlrt, level2(school: cons) level1(student: cons) nopause
```

Two-level random-intercept and random-slope (coefficient) model (fitted using IGLS)
(See page 59 of the MLwiN User Manual)
```
   . runmlwin normexam cons standlrt, level2 (school: cons standlrt) level1 (student: cons) nopause
```

Refit the model, where this time we additionally calculate the level 2 residuals (fitted using IGLS)
(See page 59 of the MLwiN User Manual)
```
   . runmlwin normexam cons standlrt, level2 (school: cons standlrt, residuals(u)) level1 (student: cons) nopause
```

Two-level random-intercept and random-slope (coefficient) model with a complex level 1 variance function (fitted using IGLS)
(See page 99 of the MLwiN User Manual)
```
   . matrix A = (1,1,0,0,0,1)
   . runmlwin normexam cons standlrt girl, level2(school: cons standlrt) level1(student: cons standlrt
girl, elements(A)) nopause
```
**runmlwin: Running MLwiN from within Stata**

**runmlwin** is a Stata command which allows Stata users to run the powerful MLwiN multilevel modelling software from within Stata.

The multilevel models fitted by **runmlwin** are often considerably faster than those fitted by the Stata’s **xtmixed**, **xtmelogit** and **xtmepoisson** commands. The range of models which can be fitted by **runmlwin** is also much wider than those commands. **runmlwin** also allows fast estimation on large data sets for many of the more complex multilevel models available through the user written **gllamm** command.

MLwiN has the following features:

1. Estimation of multilevel models for continuous, binary, count, ordered categorical and unordered categorical data
2. Fast estimation via classical and Bayesian methods
3. Estimation of multilevel models for cross-classified and multiple membership nonhierarchical data structures
4. Estimation of multilevel multivariate response models, multilevel spatial models, multilevel measurement error models and multilevel multiple imputation models

These details with a screen shot are available on our **runmlwin leaflet** (pdf, 0.1mb)
Presentations using runmlwin

- Cathie Marsh Centre for Census and Survey Research (CCSR) Seminar Series, Manchester (25th September 2012)
  - Slides (PDF, 2.0mb)
  - Stata do-file (do, 0.1mb) to replicate all analyses presented in the slides.
- 5th ESRC Research Methods Festival, Oxford (3rd July 2012)
  - Slides (PDF, 2.0mb)
  - Stata do-file (do, 0.1mb) to replicate all analyses presented in the slides.
- UK Stata Users' Group, 17th Meeting (16th September 2011)
  - Slides (PDF, 2.0mb)
  - Stata do-file (do, 0.1mb) to replicate all analyses presented in the slides.
- University of Bristol, Mplus/MlwiN User Group (MUGS) meeting (14th June 2011)
  - Slides (PDF, 2.3mb)
  - Stata do-file (do, 0.1mb) to replicate all analyses presented in the slides.
- Modern Modeling Methods (M3) Conference, University of Connecticut (26th May 2011)
  - Slides (PDF, 3.2mb)
  - Stata do-file (do, 0.1mb) to replicate all analyses presented in the slides.
- 2011 American Sociological Association Spring Methodology Conference, Tilburg University (20th May 2011)
Examples using runmlwin

MLwiN User Manual

These do-files and log files replicate the analyses reported in the MLwiN User Manual (PDF, 4.6 mb) Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2009) Centre for Multilevel Modelling, University of Bristol.

Note that we have not created do-files for Chapters 1, 8 or 19 of the manual as no models are fitted in those chapters. We have also not yet attempted to replicate the analysis in Chapter 17.

- 1 Introducing Multilevel Models
- 2 Introduction to Multilevel Modelling (do | log)
- 3 Residuals (do | log)
- 4 Random Intercept and Random Slope Models (do | log)
- 5 Graphical Procedures for Exploring the Model (do | log)
- 6 Contextual Effects (do | log)
- 7 Modelling the Variance as a Function of Explanatory Variables (do | log)
- 8 Getting Started with your Data
- 9 Logistic Models for Binary and Binomial Responses (do | log)
### ANNOUNCEMENTS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Replies</th>
<th>Views</th>
<th>Last Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>runmlwin has had 2300+ downloads since Oct 2011</td>
<td>0</td>
<td>1009</td>
<td>by GeorgeLeckie</td>
</tr>
<tr>
<td>Make sure you have latest version of runmlwin: 16/04/2012</td>
<td>0</td>
<td>1111</td>
<td>by GeorgeLeckie</td>
</tr>
<tr>
<td>Do-files to replicate entire MLwiN User &amp; MCMC Manuals</td>
<td>7</td>
<td>2131</td>
<td>by GeorgeLeckie</td>
</tr>
<tr>
<td>Welcome to the runmlwin discussion forum</td>
<td>0</td>
<td>1195</td>
<td>by GeorgeLeckie</td>
</tr>
</tbody>
</table>

### TOPICS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Replies</th>
<th>Views</th>
<th>Last Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>trouble fitting cross-classified model</td>
<td>3</td>
<td>25</td>
<td>by katehilling</td>
</tr>
<tr>
<td>error in multiple membership models</td>
<td>4</td>
<td>34</td>
<td>by morning03</td>
</tr>
<tr>
<td>Spatial multilevel models</td>
<td>2</td>
<td>13</td>
<td>by Raphael</td>
</tr>
<tr>
<td>signs of covariates change using multiple membership model</td>
<td>1</td>
<td>26</td>
<td>by GeorgeLeckie</td>
</tr>
<tr>
<td>can't proceed to multiple membership model</td>
<td>3</td>
<td>61</td>
<td>by GeorgeLeckie</td>
</tr>
<tr>
<td>Gamma Regression with random effects</td>
<td>2</td>
<td>41</td>
<td>by AndreasHaupt</td>
</tr>
<tr>
<td>Significance of random effects in cross-classified model</td>
<td>2</td>
<td>42</td>
<td>by AnjaScheewe</td>
</tr>
</tbody>
</table>
runmlwin: A Program to Run the MLwiN Multilevel Modelling Software from within Stata

George Leckie  
Centre for Multilevel Modelling  
University of Bristol

Chris Charlton  
Centre for Multilevel Modelling  
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 capitalise 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.
10. RUN MLwiN FROM WITHIN R: THE R2MLwiN FUNCTION
Continuous and binary response random slope models

• Continuous response model

\[ \text{normexam}_{ij} = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} + e_{2ij} \text{girl}_{ij} + e_{3ij} \text{boy}_{ij} \]

\[ \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \sigma_{u0} \\ \sigma_{u0} & \sigma_{u1}^2 \end{pmatrix} \right\}, \quad \begin{pmatrix} e_{2ij} \\ e_{3ij} \end{pmatrix} \sim N\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_e^2 & 0 \\ 0 & \sigma_{e3}^2 \end{pmatrix} \right\} \]

• Binary response model

\[ \text{passexam}_{ij} \sim \text{Binomial}(1, \pi_{ij}) \]

\[ \text{logit}(\pi_{ij}) = \beta_0 + \beta_1 \text{standlrt}_{ij} + \beta_2 \text{girl}_{ij} + u_{0j} + u_{1j} \text{standlrt}_{ij} \]

\[ \begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_{u1}^2 \end{pmatrix} \right\} \]
# Load the library foreign
library(foreign)

# Read in the tutorial Stata data set
tutorial = read.dta("http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial.dta")

# Generate a boy dummy variable
tutorial$boy <- 1 - tutorial$girl

# Generate the passexam binary response variable
tutorial$passexam <- tutorial$normexam > 0

# Load the library R2MLwiN
library(R2MLwiN)

# Specify the MLwiN directory
mlwin = "C:\Program Files (x86)/MLwiN v2.26/"

# Declare the levels in the model hierarchy
levID = c('school', 'student')

# Specify the continuous response model
formula = "normexam ~ (0 | cons + standlrt + girl) + (2 | cons + standlrt) + (1 | girl + boy)"

# Set the level-1 covariance between the boy and girl residual errors to zero
smat = c(2,1)

# Specify estimation by IGLS
estoptions = list(EstM = 0)

# Fit the model
mymodel = runMLwiN(formula, levID, D="Normal", tutorial, estoptions, MLwiNPath = mlwin, workdir = tempdir())
```r
# Fit the model
mymodel = runMLwin(formula, levID, D="Normal", tutorial, estoptions,
                    +
                    MLwiNPath=mlwin, workdir = tempdir())

Worksheet has 500000000 spaces
ECHO 0

Execution completed
---
MLwiN multilevel model (Normal)
Estimation algorithm: IGLS                   Elapsed time : 0.54s
Number of obs: 4059
Deviance statistic: 9281.4
---

The model formula:
normexam ~ (0 | cons + standlrt + girl) + (2 | cons + standlrt) + (1 | girl + boy)
Level 2: school     Level 1: student
---

The fixed part estimates:
                        Coef.     Std. Err.      z    Pr(>|z|) [95% Conf. Interval]
cons                    -0.11153     0.04331    -2.58     0.01001     *   -0.19641   -0.02665
standlrt                0.55294     0.02008   27.54    5.462e-167   ***   0.51359    0.59228
girl                    0.17528     0.03242    5.41    6.401e-08   ***   0.11175    0.23881
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
---

The random part estimates at the school level:
                        Coef.     Std. Err. [95% Conf. Interval]
var_cons                0.08625     0.01717      0.05259      0.11991
cov_cons_standlrt      0.01905     0.00668      0.00596      0.03214
var_standlrt           0.01489     0.00447      0.00613      0.02365
---

The random part estimates at the student level:
                        Coef.     Std. Err. [95% Conf. Interval]
var_girl               0.52516     0.01528       0.49521       0.55512
cov_girl_boy          0.00000     0.00000       0.00000       0.00000
var_boy               0.58743     0.02100       0.54628       0.62859
```

# Specify the binary response model
formula = "logit(passexam, cons) ~ (0 | cons + standlrt + girl) + (2 | cons + standlrt)"

# Specify estimation by MCMC
estoptions = list(EstM=1)

# Fit the model
mymodel = runMLwiN(formula, levID, D="Binomial", tutorial, estoptions, MLwiNPath=mlwin, workdir = tempdir())
Adapting for 1200 iterations (Maximum 5000)
Adapting for 1300 iterations (Maximum 5000)
Adapting for 1400 iterations (Maximum 5000)
Adapting for 1500 iterations (Maximum 5000)
Adapting for 1600 iterations (Maximum 5000)
Adapting finished and took 1700 iterations
Burning in for 0 iterations out of 500
Burning in for 100 iterations out of 500
Burning in for 200 iterations out of 500
Burning in for 300 iterations out of 500
Burning in for 400 iterations out of 500

Execution completed

MLwiN multilevel model (Binomial)
Estimation algorithm: MCMC Elapsed time : 28.4s
Number of obs: 4059 Number of iter.: 5000 Burn-in: 500
Bayesian Deviance Information Criterion (DIC)
Dbar D(thetabar) pb DIC 4234.584 4162.498 72.087 4306.671

The model formula:
logit(passexam,cons)-(0|cons+standlrt+girl)+(2|cons+standlrt)
Level 2: school Level 1: student

The fixed part estimates:

| Coef. | Std. Err. | z      | Pr(>|z|) | [95% Conf. Interval] | ESS |
|-------|-----------|--------|----------|----------------------|-----|
| cons  | -0.03707  | 0.10921| -0.34    | 0.7343               | -0.24130, 0.19745 | 82  |
| standlrt | 1.35860  | 0.06101| 22.27    | 7.554e-110           | 1.23754, 1.47761 | 487 |
| girl  | 0.19848   | 0.10172| 1.95     | 0.05105              | -0.00255, 0.40455 | 137 |

Signif. codes:  0 '***'  0.001 '**'  0.01 '*'  0.05 '.'  0.1 ' ' 1

The random part estimates at the school level:

<table>
<thead>
<tr>
<th>Coef.</th>
<th>Std. Err.</th>
<th>[95% Conf. Interval]</th>
<th>ESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>var_cons</td>
<td>0.51187</td>
<td>0.12163</td>
<td>939</td>
</tr>
<tr>
<td>cov_cons_standlrt</td>
<td>0.06201</td>
<td>0.05505</td>
<td>159</td>
</tr>
<tr>
<td>var_standlrt</td>
<td>0.07720</td>
<td>0.04525</td>
<td>69</td>
</tr>
</tbody>
</table>
**R2MLwiN: Running MLwiN from within R**

**R2MLwiN** is an R command interface to the MLwiN multilevel modelling software package, allowing users to fit multilevel models using MLwiN from within the R environment. It is designed to be used with versions of MLwiN from v2.25 onwards although some features will work with earlier versions.

**Installation**

Both [MLwiN](http://www.bristol.ac.uk/cmm/software/mlwin/) and [R](http://www.r-project.org/) are required to use **R2MLwiN**. [MLwiN](http://www.bristol.ac.uk/cmm/software/mlwin/) is free to UK academics. A fully functional 30-day free version of MLwiN is available to all other users.

To install **R2MLwiN**, type the following at the R command line:

```
install.packages("R2MLwiN", repos="http://cran.r-project.org")
```

**Documentation**

To see the documentation for **R2MLwiN**, type the following at the R command line:

```
library(R2MLwiN)
```
Examples using R2MLwiN

MLwiN User Manual

We provide R demos which allow you to replicate all the analysis reported in the MLwiN MCMC Manual (PDF, 7.4 mb) using R2MLwiN with the package. The table of contents are:

- 01 Introduction to MCMC Estimation and Bayesian Modelling (R)
- 02 Single Level Normal Response Modelling (R)
- 03 Variance Components Models (R)
- 04 Other Features of Variance Components Models (R)
- 05 Prior Distributions, Starting Values and Random Number Seeds (R)
- 06 Random Slopes Regression Models (R)
- 07 Using the WinBUGS Interface in MLwiN (R)
- 08 Running a Simulation Study in MLwiN (R)
- 09 Modelling Complex Variance at Level 1 / Heteroscedasticity (R)
- 10 Modelling Binary Responses (R)
- 11 Poisson Response Modelling (R)
- 12 Unordered Categorical Responses (R)