Practical sessions for ‘Introduction to multilevel models with applications’: Guide and contents

Notes for the practical sessions for the Essex Summer School 2010 course 1E, 12-23 July 2010

Paul Lambert (University of Stirling)
Version 1.10: 23/JUL/2010

Contents
Introduction ............................................................................................................................................ 2

Learning strategies .................................................................................................................................................. 2

Principles and programming .......................................................................................................................... 3

Documentation for replication .......................................................................................................................... 3

How the practical sessions are organised ........................................................................................................ 5

Example command files ................................................................................................................................... 6

Data Files ....................................................................................................................................................... 6

Macros/non-interactive command files ........................................................................................................... 7

File locations .................................................................................................................................................... 7

Analysis of your own data ................................................................................................................................ 7

The tremendous importance of workflows ...................................................................................................... 7

The Practical Programme .................................................................................................................................. 10

P1: Getting started with multilevel software and data ...................................................................................... 10
P2a/b: Examples of two-level hierarchical data/Estimating two-level multilevel models .................. 11
P3: Clustered data and variables .......................................................................................................................... 13
P4a/b: Models for random intercepts and slopes/Interpreting complex multivariate models .............. 14
P5: Analysing panel data as a multilevel dataset ............................................................................................. 15
P6a/b: Clustered data with categorical measures/Applications of models with binary outcomes . 16
P7: Data and models with complex clustering ................................................................................................. 17
P8a/b: Alternative statistical treatments for clustered data/Comparing hierarchical and other effects ........................................................................................................................................... 18
P9: Practical applications with different outcomes ............................................................................................. 19
P10: Review and recap of practical skills ............................................................................................................ 20

References ....................................................................................................................................................... 21
**Introduction**

The programme of practicals included within course 1E is intended to help you understand and interpret the topics in multilevel modelling covered in teaching sessions, and to furnish you with important skills in applying multilevel analysis techniques to social science data.

For the practical sessions we provide a single handout which briefly describes the contents of the sessions (this document) and gives some description of the Essex lab setup; a supplementary handout called our ‘How to use..’ guide which focuses upon software packages used for the labs; and extended software specific command files which contain the details of the exercises in terms of step-by-step activities for each of the practical sessions.

This guide covers the format and components of the practical sessions. At present only practicals 1 and 2 have all relevant files listed in this handout, and the component files used in the later sessions will be filled in over the course of the module. For software specific information, see the ‘How to use..’ guide.

**Learning strategies**

A lot of material is presented across the programme of practicals, particularly in the details of supplementary exercises or activities, and it is easy with software based materials to skip through parts of the sessions briefly without properly digesting them (or indeed to skip over things entirely!). Nevertheless you will get the best out of the module if you work through the exercises in full (even if that involves catching up with some aspects of them later on). You should also be prepared that covering the materials in full will often include following up on software help guides and files, over and above explicit activities from the labs – to some extent, part of learning to work with complex data and analysis methods is inevitably about teaching yourself to be a better software programmer in the area.

In addition for the sessions from week 2 we’ve prepared short ‘homework’ exercises which ought to serve to test if you have made sense of the relevant lecture and practical materials, and to reinforce your own learning on the topic.
Principles and programming

It is possible to trace three themes running through the practical sessions which summarise the approach we are adopting and the principles of good practice in social science data analysis which we follow:

- Documentation for replication
- The integration of data management and data analysis
- The use of ‘real’ data

Documentation for replication

The first principle refers to the importance of working in such a way as to keep a clear linear record of the steps involved in your analysis. This serves as a means to support the long-term preservation of your work. A useful guiding principle is that after a significant break in activities, you yourself – or another researcher at some future point – could retrace and replicate your analysis generating exactly the same results. There are several excellent recent texts which explain the importance of the principle of replicability to a model of systematic scientific enquiry with social science research data (e.g. Dale, 2006; Freese, 2007; Kohler and Kreuter, 2009). Long (2009) gives an extremely useful guide to organising you research materials for this purpose; for another comparable discussion see Altman and Franklin (2010).

In social statistics we work overwhelmingly with software packages which process datasets and generate results. In this domain, documentation for replicability is effectively achieved by (a) the careful and consistent organisation and storage of data files and other related materials, and (b) the systematic use and storage of ‘syntactical’ programming methods to process tasks.

The latter practice, syntactical programming, is not universal in the analysis of social science research data - but it should be! It involves finding a way to write out software commands in a textual format which may be easily stored, reopened, re-run and/or edited and rerun. Examples include using SPSS ‘syntax files’, and Stata ‘do files’, but almost all software programmes support some sort of equivalent syntactical command representation. A great attraction of programmes such as SPSS and Stata is that their syntactical command language is to all intents and purposes ‘human readable’. There is more on applying syntactical programming in the supplementary ‘How to use..’ guide, and we use this approach throughout the practical sessions.

The integration of data management and data analysis.

The second principle involves adequately recognising the ‘social construction’ of quantitative data in the social sciences. We try to stress through numerous examples that the way in which survey and other quantitative data is presented is ordinarily up for negotiation, such as regarding category recodes, missing data treatments, derived summary variables, and functional form choices. The way that such decisions are actually negotiated in any particular study comprises the element of research which we would label as ‘data management’.
It is our experience that any applied project requires more input to data management than to data analysis aspects of the work. As such, finding software approaches (or combinations of software approaches) which support the seamless movement back and forth between data management and data analysis is an important step forward. In this regard, Stata stands out at the current time as the superior package for social survey research as it combines extended functionally for both components of the research process (this is the main reason that Stata is used more frequently than other packages in this course). Other packages or combinations of packages can also be used effectively in this way (our other main means of working involves finding effective ways to combine SPSS with MLwiN – the former has a fairly full range of data management capacity and a good range of data analysis options; the latter has a wider range of multilevel data analysis functions but supports only limited data management tasks.

(Aside, and a shameless plug: If you are interested in the methodological topic of data management more generally, we have a related project on this very theme, where amongst other things we provide online data-oriented resources and training materials, and run workshop sessions, oriented to data management issues in social survey research: see the ‘DAMES’ project at www.dames.org.uk).

The use of ‘real’ data.

The last principle may sound rather trite but is an important part of gaining applied skills in social science data analysis. Statistics courses and texts ordinarily start with simplified illustrative data such as simulated data and data files which are very small in terms of the number of cases and variables involved. This is for very good reasons – trying anything else can lead to access permission problems, software processing time delays, and onerous data management task requirements. In this course we do often use simplified datasets which these characteristics, however as much as possible we also include materials which start from the original source data (e.g. the full version of the data such as you could get from downloading via the UK Data Archive or a similar data provider).

There are quite a few reasons why using full size data and original variable measures are important to applied research skills. Firstly, in the domain of multilevel modelling techniques, estimation and calculation problems, and the actual magnitude of hierarchical effects, then to be quite different in ‘real’ datasets compared to in simplified examples (unfortunately, in ‘real’ applications the former tend to be much greater, and the latter much smaller..!). Secondly, using full size data and original variables very often exposes problems, errors or required adjustments with software routines which may not otherwise be apparent, so it is much better to be aware of these in an original training session than only later coming across them in applied work (missing data conventions and upper limits on the numbers of categories, cases or variables are common problems here). Third, our experience is that by using ‘real’ data during practical sessions, as course participants you are much more likely to go away from the course and be readily able to apply the analytical methods covered to your own data and analysis requirements, as the data used in the sessions is much more likely to resemble the formats and structure of your own data.
How the practical sessions are organised

Materials used in this module include data files (copies of survey data used in the practicals); example command files and macros (pre-prepared materials which include programming commands in the language of the relevant software); and teaching materials (copies of presentation slides and practical session workbooks).

At each session we will give you example command files for the data analysis packages which take you through the process of carrying out the relevant data analysis operations. So, the main activities of the lab sessions involve opening the relevant files and working your way through them.

During the workshop you will be able to store data files at a network location within the University of Essex systems. You should allocate a folder for this module, and use subfolders within it to store the materials for this module.

A typical setup will look be something like:

![Folder Structure Example]

This session is from my own laptop so it won’t be identical to your machine, but you should have something broadly similar. Illustrated, I’ve made a folder called ‘d:\essex10’ within which I’ve made a number of subfolders for different component materials linked to the course. For example, in the ‘syntax’ subfolder I’ve got copies of the four example command files linked to Practical session 1; although not shown, other things from the module are in the other subfolders, such as a number of other relevant command files in the ‘sub_files’ folder, and some relevant data files in the ‘data’ folder.

A very important point to make is that the various command files all draw upon other files (e.g. data files) by referencing their network ‘paths’. For the labs to work successfully, it will be necessary to ensure that the command file you are trying to run is pointing to the right paths at the right time. In general, this only requires one specification to be made at the start of the session, for instance whereby in SPSS and in Stata we define ‘macros’ for the relevant paths (also see pages 12/20 of the ‘How to use’ guide on this issue). R and MLwiN don’t readily support the same level of programming,
however, so when you use these packages it will occasionally be necessary to edit the path reference of a particular file in order to get the right file.

For example, in the text below, we show some Stata and SPSS commands which in both cases define a macro (called ‘path3a’) which gives the directory location of the data file ‘aindresp.dta’ or ‘aindresp.sav’, so that subsequent commands calling it will go directly to that path:

<table>
<thead>
<tr>
<th><strong>Stata example</strong></th>
<th><strong>SPSS example</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>global path3a “d:\data\bhps\”</td>
<td>define !path3a () “d:\data\bhps\” !enddefine.</td>
</tr>
<tr>
<td>use pid asex using $path3a\aindresp.dta, clear</td>
<td>get file=!path3a+&quot; aindresp.sav” /keep=pid asex.</td>
</tr>
<tr>
<td>tab asex</td>
<td>fre var=asex.</td>
</tr>
</tbody>
</table>

**Example command files**

Most of the command files are annotated with descriptive comments which are intended to make them understandable to the reader. During the labs, we do of course expect that you can ask for help or clarification on any parts of the commands which are not clear.

In general, to use the example command files, you should:

- Save them into your own filestore space
- Open them with the relevant software application (e.g. the relevant syntax editor, or, alternatively, a plain text editor - see also the ‘How to use..’ document)
- Save the files with a new name to indicate your personalised copy of the file (i.e. so that you can edit the files without risk of loosing the original copies)
- Make any necessary edits to the path locations given in the files (at the top of the relevant files)
- Work through the files implementing the commands and interpreting the outputs, adding your own comments to the files to help you understand the operations you’re performing

**Data Files**

We’ll use a great many different data files over the course of the practical programme. We’re planning to supply them via the network drive, though it may sometimes be easier to transfer them by other means.

Many of these data are freely available online, such as the example data files used in textbooks such as by Hox (2010), Treiman (2009) and Rabe-Hesketh and Skrondal (2008).

Some of the data files are original versions of complex survey datasets from the UK and from comparative international studies. We can use these for teaching purposes, but you are not allowed to take them away with you from the class. You can in general access these data for free, however to do so you must register with the relevant data provider (e.g. the UK Data Archive) and agree to any associated conditions of access (e.g. required citations and acknowledgements).
Macros/non-interactive command files

As well as the example ‘interaction’ command files that you’ll open, work through and probably edit slightly, at certain points we will also supply you with command files in the various software languages which essentially serve as ‘macros’ to perform specific tasks in data management or analysis. These ordinarily do not need to be further edited or adjusted, and it is not particularly necessary to examine their contents, but it will be necessary to take some time to place the files in the right location in order that their invocation runs smoothly.

Where to put your files..

In general it will be effective to store your files (example command files, data files, and macro files) in the allocated network filestore space during the course of the summer school. This filestore should be available to you from any machine on the Essex system.

Some of the larger survey data files, however, might not be effectively stored on the network location (due to their excessive size). In some instances, it might be more efficient to store the files on the hard drives of the machine you are using (e.g. the C:\ drive), though if doing this you should be careful that contents of the hard drives may not be preserved when you log off.

The various exercises and command files ought all to be transferable to other machines (e.g. your laptop or you work machine at your home institution), though obviously you will need to install the relevant software if you don’t already have it. At the end of the module, you will probably want to take a copy of all your command files and related notes, plus the freely available example datasets and the macro files, to your memory stick or some other means, to take back with you.

Analysis of your own data

In general we would very much encourage you to bring you own data to the practical sessions. You’ll get a lot out of the module if you are able to repeat the demonstration analyses on your own data and research problems. It will ordinarily be most effective to do the class examples, then at the end of the session try to re-do it or part of it on your own data.

The tremendous importance of workflows

If you don’t already accept the tremendous importance of workflows, then this might simply be a point that, at this stage, we ask you to take in good faith! There are very good expositions of the idea of workflows in the social science data analysis process in, amongst others, Long (2009); Treiman (2009), and Kohler and Kreuter (2008).

A workflow in itself is a representation of a series of tasks which contribute to a project or activity. It can be a useful exercise to conceptualise a research project in these terms (e.g. data collection, data processing, data analysis, report writing). However, what is a really important contribution is to organise your data and command files that are associated with a project in a consistent style that recognises that relevant contributions to the workflow structure.
What does that statement mean? The issue is that we want to construct a replicable trail of our data oriented research, that allows us to go all the way from opening the initial data file, to producing the publication quality graph or statistical results that are our end product. We need the replicable trail in order to adjust our analysis on the basis of minor changes at any possible stage of the process (or to be able to transfer a record of our work on to others). However because the trail is long and complex (and not entirely linear), we can only do this, realistically, if we break down our activities into multiple separate components.

There are different ways to organise files for these purposes, but a popular and highly effective approach is to design a ‘master’ syntax command file and a series of ‘sub-files’ which it draws upon. In this model, the subfiles cover different parts of the research analysis. Personally, my preference is to construct both the master and sub-files in the principle software package being used, though Long (2009) notes that creating a documentation master file in a different software (e.g MS Excel) is an effective way to record a wider range of activities which span across different software.

Here’s an example of a series of tasks being called upon via a Stata format ‘master’ file:

(This screenshot just shows the Stata master file, and the sub-files which are mostly open within EditPad editor - except for a few other files which I’ve opened in the do file editor. The Stata output file is not visible but is obviously open behind the scenes).
Here’s an example of a project documentation file that might be constructed in Excel:

![Excel file screenshot]

Note the other tabs in the Excel file can be used to show things like author details, context of the analysis, and last update time. The file also notes some some (though not all) dependencies within the workflow – for instance step 9 requires step 4 to have been take (the macro reads in a plain text data file that was generated in Stata by do file pre_analysis1.do).

In summary, we can’t advise you strongly enough on the value of organising you data files around a workflow conceptualisation, such as through master and sub-files. Read the opening chapters of Long (2009), or the other references mentioned above, for more on this theme. Actually, we are teaching a workshop on documentation and workflows as part of the DAMES node’s workshop programme, if any of you are further interested.
The Practical Programme

P1: Getting started with multilevel software and data

<table>
<thead>
<tr>
<th>Coverage</th>
<th>A lot of things!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>➢ Get set-up with your account and file locations for the course.</td>
</tr>
<tr>
<td></td>
<td>➢ You should read every single word of the introductory guides (this one, and the ‘How to use..’ guide).</td>
</tr>
<tr>
<td></td>
<td>➢ Open up and briefly explore the examples given in each of the four packages.</td>
</tr>
<tr>
<td></td>
<td>➢ Start making your way through the Stata (or SPSS) extension file.</td>
</tr>
</tbody>
</table>

| Interactive command files | p1_stata.do  
p1_SPSS.do  
p1_mlwin.mac  
pl_R.R |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Each provides a few brief exercises in opening the packages and summarising some relevant data</td>
<td></td>
</tr>
<tr>
<td>In addition, the lab refers you to an extended introductory file covering syntax commands for the analysis of data in Stata or SPSS. These files are available at <a href="http://www.longitudinal.stir.ac.uk/workshop_materials.html">http://www.longitudinal.stir.ac.uk/workshop_materials.html</a>, called ‘lab0.do’ and ‘lab0.sps’. This site also has information on how to access the data files used in the sessions.</td>
<td></td>
</tr>
</tbody>
</table>

| Required sub-files: | The MLwiN exercise also needs: input_bhps_1.mac  
The Stata exercise draws on the online file: http://www.longitudinal.stir.ac.uk/labs/extras/uk_unis.do |

| Data files used: | [These are described in the syntax files, but you could take your own notes of the file names and/or locations] |

<table>
<thead>
<tr>
<th>Supplementary notes</th>
<th>Other comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Covers familiarity with the ways of handling data in the relevant software files, and the mechanisms for the lab session programme</td>
<td></td>
</tr>
<tr>
<td>[Don’t expect to complete all of the activities in the time allocated – but work through the background materials over the course of the week]</td>
<td></td>
</tr>
</tbody>
</table>
P2a/b: Examples of two-level hierarchical data/ Estimating two-level multilevel models

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Five examples of setting up data with a two-level hierarchical structure (lab 2a) and subsequently estimating variance components models with and without covariates for those models (lab 2b)</th>
</tr>
</thead>
</table>
| Command files | p2a_stata.do / p2b_stata  
p2a_SPSS.do / p2b_SPSS.do  
p2b_mlwin.mac  
p2b_R.R |
| The ‘2a’ files set up a number of datasets for further analysis throughout the module. The ‘2b’ files run various ‘random intercepts’ models. |
| Required sub-files: | Stata: bhps_ghq_setup_1.do; bhps_soc_setup_1.do; variance_summaries.do; from the occ_info folder: (iskoisei.ado, casoc_isco.do, soc90_labels.do, gb91soc90.dta (all at occ_info folder)  
SPSS: bhps_ghq_setup_1.sps; bhps_soc_setup_1.sps; from the occ_info folder: (iskoisei.sps, soc90_labels.sps, gb91soc90.sav (all at occ_info folder)  
MLwiN: bhps_ttwt1_input.mac; bhps_ghq_input.mac; bhps_soc_input.mac |
| Data files | [These are described in the syntax files, but you could take your own notes of the file names and/or locations] |
| Outcomes: | Understanding of the organisation of two-level hierarchical data in the relevant packages, and popular analytical approaches which do not use random effects modelling. |
| | Understanding of how to specify the principle estimation options for random intercepts models (ML, REML, GLS) via the relevant commands in Stata (xtreg, xtmixed and gllamm) or in SPSS (mixed)/MLwiN, to do this, and the typical properties and contributions of random intercepts estimates. Understanding of the interpretation of random intercepts models and of level 2 residuals. |
### Practical 2: End of practical revision test

Results of regression models or statistical analysis on a clustered dataset which ignores the clustering will tend to be *un*biased...... and *in*efficient.......(delete as appropriate).

When ‘svy;’ is used in Stata to reflect the clustering, the usual impact is to .......................... the standard errors. We say this is making the analysis more ................................. .

In Stata,
The two types of xtmixed random effects estimation options are: .............. and ............
The two types of xtreg random effects estimation options are: .............. and ............
The usual gllamm random effects estimation option is: ............................
The estimation options which will tend to give the largest ICC are: .............. and ............
The fastest estimators are in ............................... The slowest estimators are in ..........................

In SPSS,
The two types of mixed random effects estimation options are: .............. and ............
The estimation option which will tend to give the larger ICC is: .............

In MLwiN
The IGLS random effects estimation option is used to get the .............. estimator
The RIGLS random effects estimation option is used to get the .............. estimator
The estimation option which will tend to give the larger ICC is: .............

Random intercepts models model variance in the regression lines within groups as if the lines were free to ..........................................................
The role of caterpillars in multilevel modelling is ..........................................................
### P3: Clustered data and variables

| Coverage | More on techniques for describing clustered data and variables.  
|          | An example of interpreting higher versus lower level explanatory variables.  
|          | An example of centring and standardising variables.  
|          | An example showing the change (decline) in higher level random effects variance with increasing explanatory variables at either the higher or lower level.  
|          | Looking at the variance covariance matrix (Stata only).  |

| Command files | `p3_stata.do`  
|               | `p3_SPSS.sps` |

| Required sub-files | Stata: `variance_summaries.do`  
|                    | SPSS and Stata: Output data generated in labs 2a. |

| Data files | `[These are described in the syntax files]` |

| Other: | |
## P4a/b: Models for random intercepts and slopes/Interpreting complex multivariate models

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Introduction to specification of random coefficients across a variety of packages, replicating the examples discussed in lectures. Some other routines on handling and assessing regression data in the Stata file.</th>
</tr>
</thead>
</table>
| Command files | p4a_stata.do  
| | p4b_stata.do  
| | p4b_spss.sps  
| | p4b_R.R  
| | p4b_mlwin.mac  |
| Required sub-files: | For the Stata files: variance_summaries.do  
| | For the MLwiN files: bhps_ttwt_input.mac  |
| Data files | [These are described in the syntax files]  
| | Mostly files as created in session 2a  |
| Other: | A recommended exercise after looking over the examples is to generate your own models with the same datasets but trying out different variables and/or variance components.  
| | Be wary that some models which you could specify would take a long time to estimate in Stata. A rule of thumb is to give up and hit ‘break’ after 5 minutes of running, and try a different model instead. The speed differences between Stata and other packages are sometimes very substantial.  
| | A couple of bugs in the code in the SPSS file p4b_spss.sps were updated on 16/7/10 (after the lab sessions). |
### Coverage
Stata and SPSS examples which illustrate opening up the BHPS annual sample files and generating a panel dataset, then exploring and analysing long format panel data using various methods including multilevel models with random effects.

### Command files
- p5_stata.do
- p5_spss.sps

### Required sub-files:
- Stata: variance_summaries.do

### Data files
[These are described in the syntax files]
Requires access to the full BHPS sample.

### Other:
Many of the materials in these files are extracts from the exercises at www.longitudinal.stir.ac.uk; refer to those examples for some further extensions on this topic.

At time of writing (16/7/10) the SPSS syntax file would benefit from further debugging.

Since the lab session, both the SPSS and Stata files have been updated on the S drive with minor bug fixes:
- The Stata file now uses xtmixed rather than xtreg for the repeated cross-sectional model since xtreg won’t support the model with collinear predictors in Stata v11 (it’s ok in v10).
- There were several typos and erroneous cross-references within the SPSS command file which have now been adjusted.
P6a/b: Clustered data with categorical measures/Applications of models with binary outcomes

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Examine binary outcomes data in the context of hierarchical datasets; look at the implications of GLM transformations for binary outcomes; implement some standard binary outcomes models in Stata and MLwiN</th>
</tr>
</thead>
</table>
| Command files                                                           | P6a_stata.do  
P6b_stata.do  
P6a_spss.sps  
P6b_mlwin.mac                                                                                                                                  |
| Required sub-files:                                                     | variance_summaries.do (for the Stata files)  
bhps_soc2_input.mac, bhps_soc2b_input.mac, bhps_ghq2_input.mac, bhps_ghq2b_input.mac for the MLwiN exercises (the 2_input files are for reading the data generated by the 2a Stata file; the 2b_files are for reading the largely equivalent data as generated by the 2a SPSS file) |
| Data files                                                              | [These are described in the syntax files]                                                                                                                                                        |
| Other:                                                                 |                                                                                                                                                                                                  |
P7: Data and models with complex clustering

<table>
<thead>
<tr>
<th>Coverage</th>
<th>An illustration of 3 level models in SPSS, Stata, MLwiN and R, and example cross-classified model specifications in Stata, MLwiN and R.</th>
</tr>
</thead>
</table>
| Command files | p7_stata.do  
p7_spss.sps  
p7_mlwin.mac  
p7_R.R |
| Required sub-files: | For Stata: variance_summaries.do (updated 20/7/2010, on the S drive)  
For MLwiN: bhps_ghq_3level_input.mac; bhps_soc_3level_input.mac; nurses_3level_input.mac |
| Data files | [These are described in the syntax files] |
| Other: | Don’t expect the cross-classified model examples to work smoothly. I found on my laptop that both MLWiN and Stata crashed whilst trying to estimate versions of the BHPS SOC cross-classified model |
### Coverage

A few selected examples of multilevel models and other modelling approaches for summarising hierarchical patterns – mostly corresponding to the examples presented in the lecture slides (lecture 8).

### Command files

- **p8a_stata.do**  
  (Just the one pre-prepared file for this session)

### Required sub-files:

- Variance_summaries.do (as in earlier labs)

### Data files

[These are described in the syntax files]

### Other:

By this stage of the module, we’d encourage you to use the lab time to work with your own data examples as well as the pre-prepared routines.
P9: Practical applications with different outcomes

<table>
<thead>
<tr>
<th>Coverage</th>
<th>Syntax files in Stata and MLwiN show a short selection of multilevel model specifications for complex categorical data and event history data using previously prepared data extracts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command files</td>
<td>p9_stata.do</td>
</tr>
<tr>
<td></td>
<td>p9_mlwin.mac</td>
</tr>
<tr>
<td>Required sub-files:</td>
<td>For Stata: variance_summaries.do For MLwiN: hox_safety_input.mac; the p9_stata.do performs the data manipulations to generate the plain text data file accessed by the data input macro</td>
</tr>
<tr>
<td>Data files</td>
<td>[These are described in the syntax files]</td>
</tr>
<tr>
<td>Other:</td>
<td></td>
</tr>
</tbody>
</table>
P10: Review and recap of practical skills

<table>
<thead>
<tr>
<th>Coverage</th>
<th>There’s no additional materials in session 10. We suggest you use the lab time to review previous materials or to work on applying some of the multilevel models considered to your own data.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command files</td>
<td></td>
</tr>
<tr>
<td>Required sub-files:</td>
<td></td>
</tr>
<tr>
<td>Data files</td>
<td>[These are described in the syntax files]</td>
</tr>
<tr>
<td>Other:</td>
<td></td>
</tr>
</tbody>
</table>
References


