A review of random effects modelling using gllamm in Stata

(rel. 2.3.11 of 15/10/2005)

Leonardo Grilli and Carla Rampichini

Department of Statistics "G. Parenti"
University of Florence
grilli@ds.unifi.it , carla@ds.unifi.it

Contents

1. Introduction

1.1 Background

The program gllamm runs in the statistical package Stata and estimates GLLAMMs (Generalized Linear Latent And Mixed Models: Skrondal and Rabe-Hesketh, 2004) by maximum likelihood. Stata is a commercial software, while gllamm is a free program, downloadable from the web site www.gllamm.org along with the manual (Rabe-Hesketh, Skrondal and Pickles, 2004: www.bepress.com/ucbbiostat/paper160) and other useful material. A Stata book (Rabe-Hesketh and Skrondal, 2005) describes the usage of gllamm (and other Stata's commands) for multilevel and longitudinal modelling.

GLLAMM is a class of multilevel latent variable models for (multivariate) responses of mixed type including continuous responses, counts, duration/survival data, dichotomous, ordered and unordered categorical responses and rankings. The latent variables (common factors or random effects) can be assumed to be discrete or to have a multivariate normal distribution. Examples of models in this class are multilevel generalized linear models or generalized linear mixed models, multilevel factor or latent trait models, item response models, latent class models and multilevel structural equation models.

For random effects modelling, Stata has other commands for fitting specific two-level models. In particular, for panel data there is a suite of commands beginning with the prefix xt, such as xtreg for the random intercept linear model and xtlogit for the random intercept logit model. For survival data, the streg and stcox commands can estimate shared frailty models. Recently the new command xtmixed for multilevel (random coefficients) linear models with two or more levels has been included in Stata version 9.

Our review will focus on multilevel generalized linear models using gllamm. For the linear model the command xtmixed will be also used.

All models are fitted using a PC computer with Windows XP system, x86 family 2071 Mhz processor and 1024 Mb RAM.

1.2 Software and hardware requirements

Detailed information on Stata is available on the official web site www.stata.com . The latest version of Stata is 9, which is supported on many operating systems and platforms.

As for Microsoft Windows, Stata runs under a wide variety of Windows versions and on a multitude of platforms. Supported versions include Windows 2000, Windows 2003 Server and Windows XP. Also available are 64-bit versions of Stata that will allow access to greater memory allocations to handle large datasets.

Stata for Macintosh requires Mac OS X 10.1 or later, while Stata for Unix can run under Linux, IBM AIX, Sun Solaris, HP/Digital Unix and any Alpha series running Digital Unix Tru64.

Stata is available in three versions: Stata/SE, Intercooled Stata, and Small Stata. These versions differ in the size of the dataset that each one can analyze: Stata/SE is for large datasets (up to 32,766 variables, while the limit of observations is based on the amount of RAM in the computer), Intercooled Stata is the "standard" version (up to 2,047 variables, while the limit of observations is based on the amount of RAM in the computer), Small Stata is a smaller, "student" version (datasets with a maximum of 99 variables on approximately 1,000 observations).

The minimum hardware requirement are 128 MB of RAM and 60 MB of disk space.

1.3 Basics of Stata

To use `gllamm` one should know a little bit about Stata to be familiar with features common to all estimation commands including `gllamm` and to be able to prepare the data for analysis.

Stata is a general package for statistical analyses, data management, and graphics that can be used in a command-driven or menu-driven fashion. Within the software, one mainly works with four windows named Review, Variables, Results and Command, that are contained within the main Stata window. The menus and toolbar provide access to Stata's dialogs and utilities, such as the Viewer, Do-file Editor, and Data Editor.

Many resources on getting started with Stata are available on the Web. In particular, we suggest the UCLA resources to help learn and use Stata:
www.ats.ucla.edu/stat/stata/ and statcomp.ats.ucla.edu/stata/ .

Here we only briefly mention how to input/output and save the data. Any kind of input/output can be done from pop-up menus or using commands. In particular, for data input the main commands are the `infile` command, which reads unformatted ASCII data files with variables separated by spaces; or the `insheet` command which reads comma separated files and tab separated file (Stata examines the file and determines whether commas or tabs are being used as separators and reads the file appropriately). Variable names and missing values coded as dot or other symbols can be read in without problems. Fixed format files can be read with the `infix` command.

The `edit` command opens the Data Editor, which resembles an Excel spreadsheet and can be used to insert data directly. This kind of input is useful when the data are on paper and need to be typed in.

One can view the data using the `browse` command. Stata works with only one data file at the same time. For data output, the `save` command can be used to save files in Stata's .dta format. Using the Export Tool in the File list, one can output data in standard data formats with space, comma or tab delimiters. Results in the form of texts or tables which appear in the Results window can be copied/cut and pasted into a word processor file or saved in a log ASCII file using the `log` command.

A statistical task such as model fitting can be carried out through commands. A Stata *command* is a *statement* that can be followed by many *options*. Most of Stata's commands share a common syntax, which is:

```
    [prefix_cmd:] command [varlist] [if] [in] [, options]
```

where items enclosed in square brackets are optional.

This review will show the syntax for fitting a variety of random effect models using mainly the `gllamm` command.

Programming statements are available in Stata which enables complex data processing or data management. In this review, we use only the necessary basic commands in order to prepare the data for fitting specific random effect models.

1.4 The **gllamm** *program*

The gllamm program runs within Stata 6, 7, 8 and 9 using a similar syntax to Stata's own estimation commands. After estimating a model using gllamm, the command gllapred can be used to obtain the posterior means and standard deviations of the latent variables (random effects) and other predictions. The command gllasim can be used to simulate from the model.

The full syntax of the gllamm command with all available options is

```
gllamm depvar [varlist] [if exp] [in range] , i(varlist)
[noconstant offset(varname) nrf(#,...,#) eqs(eqnames)
frload(#,...,#) ip(string) nip(#,...,#)peqs(eqname) bmatrix(matrix)
geqs(eqnames) nocorrel constraints(clist)weight(varname)
pweight(varname) family(familynames) fv(varname) denom(varname)
s(eqname) link(links) lv(varname) expanded(varname varname string)
basecategory(#) composite(varname varname...) thresh(eqnames)
ethresh(eqnames) from(matrix) copy skip long lf0(# #) gateaux(# # #)
search(#) noest eval init iterate(#) adoonly adapt robust
cluster(varname)level(#) eform allc trace nolog nodisplay dots ]
```

The options are fully described in the gllamm manual (Rabe-Hesketh, Skrondal and Pickles, 2004) and in the appendix of Rabe-Hesketh and Skrondal (2005).

The syntax of the command reflects the structure of the GLLAMM class of models (Skrondal and Rabe-Hesketh, 2004), whose components are:

1. the conditional expectation of the responses given the latent and observed explanatory variables, where random effects are just an instance of latent variables ;
2. the conditional distribution(s) of the responses given the latent and observed explanatory variables;
3. structural equations for the latent variables including regressions of latent variables on explanatory variables and regressions of latent variables on other latent variables;
4. the distributions of the latent variables.

In this review we consider a subset of GLLAMMs that doesn’t require the specification of point 3.

gllamm maximises the marginal log-likelihood using Stata's version of the Newton Raphson algorithm. In the case of discrete random effects, the marginal log-likelihood is evaluated exactly, whereas numerical integration is used for continuous (multivariate) normal random effects. Various types of quadrature are available for numerical integration: the rule for determining the points of integration can be cartesian or spherical, while the procedure can be ordinary or adaptive. In any case *it is essential to make sure that a sufficient number of quadrature points has been used* by comparing solutions with a different number of quadrature points. In most cases adaptive quadrature will perform better than ordinary quadrature. This is particularly the case if the cluster sizes are large and the responses include (large) counts and/or continuous variables. Even where ordinary quadrature performs well, adaptive quadrature often requires fewer quadrature points making it faster.

For simple problems, gllamm is usually easy to use and does not take a very long time to run. However, the program can be very slow when there are many latent variables in the model, many quadrature or free mass-points, many parameters to be estimated and many observations. The reason for this is that numerical integration is used to evaluate the marginal log-likelihood and numerical derivatives are used to maximize it. Roughly, execution time is proportional to the number of observations and the square of the number of parameters. For quadrature, the time is approximately proportional to the product of the number of quadrature points for all latent variables used. For example, if there are two random effects at level 2 (a random intercept and slope) and 8 quadrature points are used for each random effect, the time will be approximately proportional to 64. Therefore, using 4 quadrature points for each random effect will take only about a quarter (64/16) as long as using 8. For (2-level) discrete latent variables, the time is proportional to the number of points, but the increase in the number of parameters must be taken into account. For details on the computational aspects refer to Rabe-Hesketh, Skrondal and Pickles (2002) and Rabe-Hesketh, Skrondal and Pickles (2005).

2. Model specifications —Basic models

In this section, we explore some basic multilevel models that can be fitted using `gllamm`: Normal models, logit/probit models for binary data and Poisson models for count data. We will describe the syntax needed for specifying the models, as well as the estimates and computing times.

2.1 Two-level Normal models

In general it is not advisable using `gllamm` for normally distributed responses since plenty of software exists for fitting such models without using approximations such as quadrature. However, if `gllamm` is used, adaptive quadrature is likely to give better parameter estimates than ordinary quadrature. With both methods, the user must ensure that sufficient quadrature points are used.

Although adaptive quadrature is likely to give good estimates for continuous responses as long as enough quadrature points are used, it is certainly more computationally efficient to use software that does not use any approximations for this particular case, e.g. the command `xtmixed` of Stata.

The data set to be used is the example which appears in the user's guide to MLwiN (Rasbash *et al.* 2005). It consists of 4,059 students (level 1 units) nested within 65 schools (level 2 units). The outcome is their examination score at age 16 (`EXAM`). A key covariate is the prior London Reading Test score (`STANDLRT`) taken at age 11. Both the outcome and the reading scores were standardized with zero mean and unit variance and in addition the outcome score was normalized. Another student level variable is gender (`GENDER`: code 1 for girls and 0 for boys). Also considered is the school level variable concerning the school gender (`SCHGEND`, coded 1 for mixed schools, 2 for boys schools and 3 for girls schools). Before fitting the models, the data are assumed to have been read into Stata and the data file is named `exam.dta`.

Five models are fitted, each one an extension of another.

2.1.1 Model A

Model A is a variance components model with fixed effects for all three covariates. For the categorical variable `SCHGEND` (school gender), two dummies are put into the model with girls schools as the reference category. Only the intercept is allowed to have random effects u_{0j} among schools. The model can be written as

$$\begin{aligned} y_{ij} &= \beta_{0ij} + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3j} + \beta_4 x_{4j} \qquad (1) \\ \beta_{0ij} &= \beta_0 + u_{0j} + e_{0ij} \end{aligned}$$

where the errors are assumed to be independent with distributions

$$u_{0j} \sim N(0,\sigma^2_{u_0}) \quad e_{0ij} \sim N(0,\sigma^2_{e_0})$$

The β s in (1) are fixed effects, and $\sigma^2_{u_0}$ and $\sigma^2_{e_0}$ are two variance parameters to be estimated. This model is used to show the usage of the basic statements in the commands gllamm and xtmixed for fitting a simple two-level linear model.

In terms of the components of a GLLAMM, in model A the conditional distribution is Gaussian, the link is identity and the marginal distribution of the random effects is Gaussian. These specifications are the default in gllamm and the only choice in xtmixed. By default, both gllamm and xtmixed assume a random intercept model.

Table 1 reports the results for Model A obtained using four different estimation methods: xtmixed REML; xtmixed ML via the EM algorithm; gllamm ML via adaptive quadrature; gllamm nonparametric maximum likelihood (NPML).

The gllamm syntax for fitting model A with ML via adaptive quadrature is

```
xi: gllamm exam standlrt gender i.schgend, i(schid) adapt nip(8)
```

The first name after gllamm defines the response variable, while the subsequent names (until the comma) refer to the covariates.

The command prefix xi: declares that Stata will have to produce dummy variables for categorical covariates present in the model. Categorical covariates are specified by putting the prefix i. before the variable name. For every categorical variable with integer codes 1,2,…,*g* appearing in the independent variables list, *g*-1 dummy variables will be included. The default reference group of the command prefix xi: is the first code, i.e. code 0 (boys) for the individual gender and code 1 (mixed school) for the school gender in our case. One can change the default reference category by means of the char command.

The i() option gives the variable that identifies the clusters.

By default gllamm performs ML with 8-point ordinary Cartesian (Gauss-Hermite) quadrature. The option adapt requests adaptive quadrature to be used instead of ordinary quadrature. The nip() option specifies the number of integration points: we used the default 8 points, obtaining the same results as ML through the command xtmixed, even if the computing time is much longer.

2.1.2 Model A via nonparametric maximum likelihood

To perform ML the marginal distribution of the random effects u_{0j} is assumed to be Gaussian. Alternatively, it can be assumed a discrete distribution with non-zero probability on a finite number of points. The discrete distribution can be interpreted as representing a number of latent classes which are homogeneous in the unobserved characteristic represented by the latent variable, e.g. in their intercepts.

If the number of points, or masses, is chosen to achieve the largest possible likelihood, the nonparametric maximum likelihood estimator (NPML) can be achieved (Lindsay, 1995; Rabe-Hesketh, Pickles and Skrondal, 2003). The Gateaux derivative method is used to determine the number of masses required for the NPML solution. See the gllamm manual for further details and references.

The gllamm option for fitting a model using a discrete distributions for the random effects is ip(f). In such a case the option nip() gives the number of mass points. A constraint on the mass points is imposed by default in order to have a distribution with zero mean.

The NPML estimates of model A are obtained with an iterative procedure:

1. a model with 2 mass points is fitted:

   ```
   xi: gllamm exam standlrt gender i.schgend, i(schid) ip(f) nip(2)
   ```

2. the results of the model are saved and passed to a new `gllamm` command which computes the Gateaux derivatives

   ```
   matrix b=e(b)
   local ll=e(ll)
   local k=e(k)
   xi: gllamm exam standlrt gender i.schgend, i(schid) ip(f) nip(3)/*
        */ from(b) gateaux(-2, 2, 20) lf0(`k' `ll')
   ```

3. If the maximum Gateaux derivative is not positive the program stops, otherwise the same `gllamm` command goes on and fits a further model with an additional mass point using the estimates of the previous step as initial values;

4. Step 2 and 3 are repeated until the maximum Gateaux derivative is not positive.

The `gateaux(min max n)` option is used with method `ip(f)` to increase the number of mass points by one from a previous solution with parameter estimates specified using the `from()` option and number of parameters and log-likelihood specified by `lf0(# #)`. The program searches for the location of the new mass point by placing a very small mass at the location given by the first argument and moving it to the second argument in the number of steps specified by the third argument. The boundaries of the grid for the `gateaux` search can be chosen by looking at the estimated standard deviation of the random effect (in this case we have chosen the values ±2 as 2 is about 6 times the estimated standard deviation of u_{0j}). If the maximum increase in likelihood is greater than 0, the location corresponding to this maximum is used as the initial value of the new location, otherwise the program stops. When this happens, it can be shown that for certain models the current solution represents the nonparametric maximum likelihood (NPML) estimate.

However, before assuming that the NPML estimator has been found, it is advisable to change the grid being searched. In our application, we tried several grids. With some grids, e.g. `gateaux(–2, 2, 100)`, the process does not end because of a negative derivative, but since a computational error occurs during the five-point fitting. This could be due to the starting values since these will change a bit when changing the grid. In such cases, instead of retaining the four-point estimates, it is advisable to try other grids until the search stops ‘regularly’ due to a negative derivative.

With other grids, e.g. `gateaux(-2, 2, 20)` and `gateaux(-3, 3, 100)`, the process ends ‘regularly’ with five mass-points. This appears to be the NPML estimator since a variety of grids all lead to the conclusion that no more masses should be introduced (and the final locations and probabilities are exactly the same).

The results are reported in Table 1.

Note that the discrete distribution of the random effects is quite symmetric around zero and has a variance similar to the one obtained with ML. The fixed parameters for individual level covariates are nearly identical to those previously obtained, while the estimates for the school level variables are slightly different.

2.1.2 Models B and C

The following extensions B and C of model A could be fitted with `gllamm`. However here we show the REML estimates obtained with the command `xtmixed`, for reasons of computational efficiency and comparability with other software for linear mixed models. The results are reported in Table 2.

Model B is an extension of Model A including an interaction between the two student level variables, London Reading Score and student gender. This illustrates the ease with which the prefix `xi:` allows to generate interactions (this is possible for any Stata command for fitting models).

Model C extends Model B further by allowing the effect of the prior student's achievement β_1 to vary randomly among the population of schools:

$$y_{ij} = \beta_{0ij} + \beta_{1j} x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3j} + \beta_4 x_{4j} + \beta_5 (x_{1ij} \times x_{2ij}) \qquad (2)$$

$$\beta_{0ij} = \beta_0 + u_{0j} + e_{0ij}$$

$$\beta_{1j} = \beta_1 + u_{1j}$$

The random effects at the school level u_{0j} and u_{1j} are assumed to follow a bivariate Normal distribution with zero mean and unconstrained covariance matrix. This type of covariance structure is termed *Unstructured* (UN) in `xtmixed`, and it is requested by the option `cov(unstruct).` Other covariance structures of the random effects are available in `xtmixed`: (i) independent (the default): one variance parameter per random effect, all covariances zero; (ii) exchangeable: equal variances for random effects, and one common pairwise covariance; (iii) identity: equal variances for random effects, all covariances zero.

2.1.2 Models D and E

Models D and E extends Model C by allowing a heteroscedastic residual variance at level 1. Such models cannot be fitted with `xtmixed`, but `gllamm` can do the job.

In Model D the level 1 variance is allowed to be different for boys and girls using the following syntax:

```
eq sch_c: cons
eq sch_lrt: standlrt
gen boy=(gender==0)
gen girl=(gender==1)
eq vargen: boy girl
xi: gllamm exam  i.gender*standlrt i.schgend , i(schid) nrf(2)/*
    */ eqs(sch_c sch_lrt) s(vargen) ip(m) adapt nip(7)
```

`boy` and `girl` are dummy variables for the gender. The `eqs()` option of `gllamm` allows fitting of random slope models, either structured or unstructured. `eqs(eqnames)` specifies the equation names (defined before running `gllamm`) for the linear predictors of the random effects. The number of random effects is specified in the `nrf()` option.

The `s(eqname)` option allows to specify an equation for level 1 heteroscedasticity since the log of the standard deviation at level 1 is modeled by the linear predictor defined by `eqname`.

The ip(m) option requests spherical quadrature (Rabe-Hesketh, Skrondal and Pickles, 2005), which is computationally more efficient for multidimensional integration.

The post-estimation command nlcom allows to calculate the estimated values and standard errors (via the delta method) of nonlinear transformations of model parameters. In this case we used it to obtain the level one variances:

```
matrix list e(b)

nlcom (s_boy: exp(2*[lns1]boy))  (s_girl: exp(2*[lns1]girl))
```

The command matrix list e(b) allows to see how gllamm names the parameters.

Model E is another version of Model D. Instead of allowing the residual variance to differ between the genders, it models the residual variance as an exponential function of the student intake variable as follows

$$\mathrm{var}(e_{ij}) = \sigma_e^2 \exp(x_{1ij}\delta) \tag{3}$$

where σ_e^2 and δ are parameters to be estimated.

The syntax and model estimates, as well as convergence times are listed in Table 3.

Since gllamm specifies the linear predictor for the log of the standard deviation at level 1, the post-estimation command:

```
nlcom (s_e: exp(2*[lns1]const )) (delta: 2*[lns1]standlrt)
```

allows one to recover $\hat{\sigma}_e^2$ and $\hat{\delta}$, along with their standard errors (via the delta method).

2.2 Other useful features of gllamm

gllamm is a Stata estimation command and thus shares the Stata facilities associated with estimation commands. For example:

- To carry out custom hypothesis tests on any fixed parameters or random parameters, the post-estimation commands lincom (for linear combinations) and nlcom (for nonlinear combinations) are available.
- To evaluate overall goodness of fit and compare nested models, one can save model results via the estimates store name command (where name is user-chosen name for the model) and then use the post-estimation command estimates stats that gives log-likelihood, AIC and BIC values. To compare the parameter estimates of different models one can use the post-estimation command estimates table.

Other useful features of gllamm are the following:

- Linear constraints on model parameters at any level can be placed using the option constraint(clist), where clist specifies the constraint numbers of the linear constraints to be applied. Constraints are defined using the constraint command. To find out the equation names needed to specify the constraint one can run gllamm with the noest and trace options. For example, in model A we can fix the first level variance to 1 and we can impose the equality between the coefficients of the two dummies corresponding to schgend:

```
constraint define 1 [lns1]_cons=0

constraint define 2 _Ischgend_1 =_Ischgend_2
```

```
xi: gllamm exam standlrt gender i.schgend, i(schid) /*
       */ constraint(1 2) adapt
```

- To obtain the empirical Bayes prediction of level 2 residuals (random effects) after the command `gllamm`, one can use the command `gllapred` with the `u` option. Together with each residual prediction, the posterior standard deviation (also called *comparative* standard error) is reported. Using the option `ustd` produces standardized posterior means, i.e. estimated residuals divided by their sampling standard deviations (also called *diagnostic* standard errors).
- Diagnostic measures can be obtained by the command `gllapred` with the options `cooksd` (Cook's distances for the top-level units), `pearson` (Pearson residuals), `deviance` (deviance residuals) and `anscombe` (Anscombe residuals).
- The `robust` option specifies that the Huber/White/sandwich estimator of the covariance matrix of the parameter estimates is to be used. If a model has been estimated without the `robust` option, the robust standard errors can be obtained by simply typing `gllamm, robust`. Moreover, the `cluster(varname)` option specifies that the highest level units of the model are nested in even higher level clusters where `varname` contains the cluster identifier. Robust standard errors will be provided that take this clustering into account. If a model has been estimated without this option, the robust standard errors for clustered data can be obtained using the command `gllamm, cluster(varname)`.
- Two kinds of weight are allowed. (1) The option `weight(wt)` specifies that variables `wt1`, `wt2`, etc., contain *frequency weights*. The suffixes in the variable names determine at what level each weight applies (if only some of the weight variables exist, e.g. only level 2 weights, the other weights are assumed to be equal to 1). If many observations with the same pattern are present in the data set, the use of frequency weights allows to reduce the dimension of the data set in a substantial manner, speeding up the computation. (2) The `pweight(varname)` option specifies that variables `varname1`, `varname2`, etc. contain *probability weights* for levels 1, 2, etc. As far as the estimates and log-likelihood are concerned, the effect of specifying these weights is the same as for frequency weights, but the standard errors will be different. Robust standard errors will automatically be provided. The use of probability weights is discussed in Rabe-Hesketh and Skrondal (2006).

2.2 Two-level models for binary/binomial data

To illustrate how `gllamm` works with binary/binomial responses, we fitted models to data from the 1989 Bangladesh Fertility Survey (Huq and Cleland, 1990). The data are a sub-sample of 1934 women grouped into 60 districts. The outcome variable is use of contraception (`use`) which equals 1 for using contraception and 0 otherwise. Three covariates are considered: age at survey centered at the sample mean (`age`); type of region of residence (`urban`) which equals 1 for urban and 0 for rural; and number of living children (`nchild`: 1=none, 2=one, 3=two, 4=three or more), represented by three dummy variables for the last three categories. The random intercept two-level model is:

$$\begin{aligned} & y_{ij} \mid \mathbf{x}_{ij}, u_{0j} \sim B(1, \pi_{ij}) \\ & g(\pi_{ij}) = \beta_0 + \boldsymbol{\beta}\mathbf{x}_{ij} + u_{0j}, \quad u_{0j} \sim N(0, \sigma_u^2) \end{aligned} \tag{4}$$

where $g(.)$ is the link function.

We first fitted model (4), using both logit and probit link functions. The gllamm syntax for fitting the logit model is

```
xi: gllamm use urban age i.nchild, i(district) family(binomial)/*
        */ link(logit) nip(8) adapt
```

To fit the corresponding probit model it is sufficient to replace the link with the option link(probit).

We then fitted a model which allows both a random intercept and a random slope for the effect associated with the type of region of residence. The gllamm syntax for fitting this model with logit link and adaptive spherical quadrature is:

```
eq urban: urban
eq cons: cons
xi: gllamm use urban age i.nchild, i(district) family(binomial) /*
        */ link(logit) nrf(2) eqs(cons urban) ip(m) nip(7) adapt
```

The results are reported in Table 4. gllamm yields the same results as other packages using ML via the quadrature integration method.

For binomial data where the outcome is a proportion *p* based on *n* observations for each case, the option denom(varname) gives the variable containing the binomial denominator for the response. The default denominator is 1.

2.3 A two-level model for count data

For hierarchical count data, Poisson models can be fitted using gllamm. Here we illustrate the use of gllamm on the malignant melanoma mortality data from 354 counties within 78 regions within 9 European countries. The data consist of observed deaths and expected deaths due to malignant melanoma, which produce the standardised mortality rate (SMR), (*observed deaths*)/(*expected Deaths*). One important environmental variable that might be associated with the mortality rate is the county level UV radiation exposure. A detailed analysis is presented in Langford, Bentham and McDonald (1998). Due to the small number of countries, we initially treat the data as having a two-level structure with counties nested within regions.

Let y_{ij} be the observed death count in the i^{th} county from the j^{th} region which is assumed to follow a Poisson distribution with mean λ_{ij}. Denoting with E_{ij} the expected death count, it follows the standardised mortality rate $SMR_{ij}=y_{ij}/E_{ij}$.

A simple variance components model for *SMR* is:

$$SMR_{ij} = \exp(\beta_0 + u_{0j} + \beta_1 x_{ij}),\ u_{0j} \sim N(0, \sigma_u^2), \tag{5}$$

where x_{ij} is the UV exposure measure, and u_{0j} are the random effects for the regions. Alternatively, model (5) can be written as:

$$\begin{aligned} &y_{ij} \mid x_{ij}, u_{0j} \sim Poisson(\lambda_{ij}) \\ &\log \lambda_{ij} = \log E_{ij} + \beta_0 + \beta_1 x_{ij} + u_{0j} \end{aligned} \tag{6}$$

where $\log E_{ij}$ is an offset. We generate this variable with the command:

```
    gen lne=log(nexp)
```

The gllamm syntax for fitting model (6) is:

```
    gllamm nobs uvb, i(idregion) offset(lne) family(poisson) /*
               */ link(log) nip(8) adapt
```

The results are reported in Table 5. Random slope models can be fitted by extending the syntax as in the linear case (see Table 3).

In order to detect overdispersion we include in the model a random intercept at level one ζ_{ij}~N(0,ψ^2), as suggested in Rabe-Hesketh and Skrondal (2005, §6.7.1):

$$\log(\lambda_{ij}) = \log E_{ij} + \beta_0 + u_{0j} + \beta_1 x_{ij} + \zeta_{ij} \tag{7}$$

The mean and variance of the counts given the covariates and the cluster random effect, but marginally w.r.t. the random intercept at level one, are:

$$\lambda_{ij}^M = E(y_{ij} \mid x_{ij}, u_{0j}) = \exp\left(\log E_{ij} + \beta_0 + u_{0j} + \beta_1 x_{ij} + \frac{\psi^2}{2} \right)$$

$$Var(y_{ij} \mid x_{ij}, u_{0j}) = \lambda_{ij}^M + \left(\lambda_{ij}^M\right)^2 \left\{\exp(\psi^2) - 1\right\}$$

Therefore the variance of the random intercept at level one, ψ^2, can be interpreted as an overdispersion parameter.

The gllamm command for the model with overdispersion (7) is obtained by adding the idcounty variable in the i() option:

```
    gllamm nobs uvb, i(idcounty idregion) offset(lne) family(poisson)/*
               */link(log) nip(8) adapt
```

The LRT test comparing the models with and without overdispersion gives a χ^2 statistic of 17.77 that is highly significant.

We also estimate a three-level model (counties within regions within nations) by adding the nation variable in the i() option:

```
    gllamm nobs uvb, i(idregion nation) offset(lne) family(poisson) /*
               */ link(log) nip(10) adapt
```

To achieve convergence 10 quadrature points are needed. The results are reported in Table 5.

The LR test for the variance at the nation level is significant. Moreover, w.r.t. the two-level model the variance at the region level is substantially reduced.

In order to test for overdispersion, we re-fit the previous model with the addition of the idcounty variable in the i() option. The LR test for overdispersion is still significant (χ^2 statistic =17.27).

3. Model specifications —other random effects models

In this section we describe how to fit random effects model for multinomial (ordered and unordered) responses, bivariate continuous responses and bivariate binary/continuous resposes. An application to meta-analysis is also shown.

3.1 Random effects models for a multiple categorical response

In this section we consider ordinal multinomial models for ordered responses and nominal multinomial models for unordered responses.

As an example, a sub-sample of the British Social Attitudes Survey, a panel study, is considered (McGrath and Waterton, 1986). The data consist of 264 adult respondents in 54 districts who completed interviews in 1983, 1984, 1985 and 1986. The outcome of interest is a total score, ranging from 1 to 7, calculated as the number of positive answers to seven questions about whether they supported or opposed a woman's right to have an abortion under different circumstances. The score is arranged to reflect attitudes from the most restrictive (code 1) to the most permissive (code 7) towards to a woman's right to have an abortion. The data are structured as years nested within respondents within districts. There were many covariates measuring the background and socio-economic status as well as the political party preference of the respondent. For the purpose of illustrations, we consider only the religion of the respondent: Roman Catholic=1, protestant/Church of England=2, others=3 and none=4 for illustration purpose. Moreover, for comparability with the reviews of other packages, we fit a two-level model ignoring the district level.

3.1.1 Two-level random effects proportional odds model

Let us assume that, conditional on the random effects, the seven response categories follow a multinomial distribution, with cumulative conditional probabilities:

$$\gamma_{ij}^{s} = P(y_{ij} \leq s \,|\, u_j) \qquad \left(\sum_{s=1}^{7} \gamma_{ij}^{(s)} = 1\right) \tag{8}$$

for the i^{th} year of the j^{th} subject.

The simplest model for ordinal data is the proportional odds model with fixed effects for the three religion groups (`religion`) contrasting to Roman Catholic, and random effects (u_j) for respondents. The model can be written as:

$$y_{ij}^{(s)} = \log\left(\frac{\gamma_{ij}^{(s)}}{1-\gamma_{ij}^{(s)}}\right) = \alpha^{(s)} - \left(\sum_{l=1}^{3} \beta_l x_{lij} + u_j\right), \qquad u_j \sim N(0, \sigma_u^2) \tag{9}$$

The six thresholds $\alpha^{(s)}$ will have values between –∞ and +∞ on the logit scale.

The proportional odds model may be specified in `gllamm` using the link `ologit`:

```
    xi: gllamm score i.religion, i(id) family(binomial) link(ologit) /*
                */ nip(8) adapt
```

The results are reported in Table 6.

3.1.2 Two-level random effects multinomial model

Suppose that the response is an unordered categorical variable (s=1,…,7), and we fit a multinomial model with a series of odds ratios comparing each of s-1 categories with a base category, s=7 in our case. The simplest model with common random effects for each odds ratio would be

$$\log\left(\frac{p_{ij}^{(s)}}{p_{ij}^{(7)}}\right)=\beta_0^{(s)}+u_j+\sum_{l=1}^{3}\beta_l^{(s)}x_{lij}^{(s)}\ ,\ u_j\sim N(0,\sigma_u^2) \tag{10}$$

$$p_{ij}^{(s)}=\exp\left(\beta_0^{(s)}+u_j+\sum_{l=1}^{3}\beta_l^{(s)}x_{lij}^{(s)}\right)\times\left(1+\sum_{h=1}^{6}\exp\left(\beta_0^{(h)}+u_j+\sum_{l=1}^{3}\beta_l^{(h)}x_{lij}^{(h)}\right)\right)^{-1}$$

For simplicity, we will ignore the covariates in the fixed part of the model. The syntax for this model is as follows:

```
gllamm score, i(id) base(7) link(mlogit) family(binom) nip(8) adapt
```

ML estimates for this model are given in Table 6. For aggregated data in the form of proportions, the `weight` option can be used to specify the model.

3.2 Multivariate Normal response model

Modelling multiple responses, accounting for the dependence between them, is of interest to many researchers. By treating the responses as repeated measurements nested within subject, the multivariate model fits neatly into the multilevel framework. The example consists of scores on a science examination obtained on both a traditional written paper (y_1) and coursework (y_2) from 1905 16 years old students in 73 schools in England. The data have a three-level structure of exam marks nested within students within schools. Interest in these data centres on the relationship between the two components at both school and student levels, whether there are gender (1 for girls and 2 for boys) differences in this relationship and whether the variability differs for the two components. The bivariate Normal model was considered for this purpose.

Let i=1,2 indicate the responses at level 1, j the students at level 2, and k the schools at level 3. The model to be fitted is as follows:

$$\begin{cases} y_{1jk}=\alpha_1+\beta_1 x_{jk}+v_{1k}+\varepsilon_{1jk} \\ y_{2jk}=\alpha_2+\beta_2 x_{jk}+v_{2k}+\varepsilon_{2jk} \end{cases} \tag{11}$$

$$\begin{pmatrix} v_{1k} \\ v_{2k} \end{pmatrix}\sim N(0,\Omega_v):\ \Omega_v=\begin{pmatrix} \sigma_{v_1}^2 & \\ \sigma_{v_1v_2} & \sigma_{v_2}^2 \end{pmatrix}$$

$$\begin{pmatrix} \varepsilon_{1jk} \\ \varepsilon_{2jk} \end{pmatrix}\sim N(0,\Omega_\varepsilon):\ \Omega_\varepsilon=\begin{pmatrix} \sigma_{\varepsilon_1}^2 & \\ \sigma_{\varepsilon_1\varepsilon_2} & \sigma_{\varepsilon_2}^2 \end{pmatrix}$$

To fit into the GLLAMM framework the model must be written as a single equation:

$$y_{ijk}=d\,y_{1jk}+(1-d)y_{2jk} \tag{12}$$

where the indicator d is 1 for written paper and 0 for coursework.

In this model, β_1 and β_2 are the gender effects on written paper and coursework respectively. At the school level, the covariance matrix Ω_v represents the variability of marks of the two components among schools, with the correlation coefficient calculated as $\rho_v = \sigma_{v_{12}} / \sigma_{v_1}\sigma_{v_2}$. Similarly, the covariance matrix Ω_ε refers to the student level.

In order to fit the model with `gllamm` the data must be prepared appropriately:

(1) In the dataset the missing values are coded as -1, while the Stata code for missing values is a full stop. Therefore the missing values are recoded as follows:

```
replace y1=. if y1==-1
replace y2=. if y2==-1
```

(2) For the `reshape` command in the next step, the student identification code should be unique, no matter which school the student comes from:

```
egen id = group(school student)
```

(3) The dataset should be structured as a three-level hierarchy with the responses stacked in one column. To this end the response variable names must have the same prefix followed by successive integers (e.g. `y1` and `y2`) and the new unique response must have a name corresponding to the common prefix (e.g. `y`):

```
reshape long y, i(id) j(index)
```

the `reshape` command creates a new variable, here called `index`, containing the response index (here `index` is 1 if `y` contains `y1` and 2 if `y` contains `y2`).

(4) It is necessary to create dummy variables for the variable `index` as well as interaction terms for the set of dummies and every covariate (here `gender)` with the following commands:

```
generate index1=(index==1)
generate index2=(index==2)
generate gender1=gender*index1
generate gender2=gender*index2
```

The `gllamm` syntax for fitting model (12) is the following:

```
eq index1: index1
eq index2: index2
constraint define 1 [lns1]_cons=1.5
gllamm y index1 index2 gender1 gender2, nocons i(id school) /*
        */ nrf(2 2)  eqs(index1 index2 index1 index2) /*
        */ constraint(1) nip(8) adapt
```

The eq commands define equations for the random effects, while the constraint command fixes the logarithm of the level one standard deviation to a given value. Since the level one variance should be null, the log of the level one standard deviation should be fixed to a very low value, but this leads to computational problems. A pragmatic solution is to fix the log of the level one standard deviation in such a way the corresponding variance is far less than both variances at the student level, but not too small (the variances at the student level are estimated by fitting univariate two-level models: in this case the estimates are 124.86 for written paper and 179.96 for coursework). In this application we have chosen the value 1.5, corresponding to a variance of 20.09. Such a value must be added to the estimated variances of the random effects at the student level to recover the correct estimates.

The option `nocons` prevents `gllamm` from inserting the overall constant in the model.

To ease estimation it is advisable to pass initial values to `gllamm`. Such values can be obtained by fitting separate models for each response, using two distinct `xtmixed` commands (which run very fast, but then passing the estimates to `gllamm` is somewhat tricky); otherwise, one can run a single `gllamm` command with the same syntax as before and the addition of the `nocorr` option (this option requests non correlated random effects at all levels).

Table 7 shows the estimates obtained with adaptive spherical quadrature using two distinct degrees of approximation 7 and 15: a comparison with the estimates produced by other programs reveals that 7 points are enough for the estimation of the variances, but the covariances are quite underestimated. With 15 points also the covariances are well estimated, even if the computational time increases dramatically.

However, a good precision with a short computational time can be obtained if the the number of random effects is reduced by decomposing the student level errors in the following way: $\varepsilon_{1j}=u_j+e_{1j}$ and $\varepsilon_{2j}=\lambda u_j+e_{2j}$, where e_{1j} and e_{2j} are i.i.d. Normal with zero mean and variance θ^2, the factor u_j is Normal with zero mean and variance φ^2, and λ is a factor loading (with the loading on the first equation fixed to one). Therefore, $\text{var}(\varepsilon_1)=\varphi^2+\theta^2$, $\text{var}(\varepsilon_2)=\lambda^2\varphi^2+\theta^2$, $\text{cov}(\varepsilon_1,\varepsilon_2)=\lambda\varphi^2$, so the variance-covariance parameters at the student level can be easily recovered from the estimates of θ^2,φ^2, and λ. At the school level the model is specified as before. The gain in computational time (on a PC with 2071 Mhz processor) is substantial: half an hour versus about two days!

The `gllamm` command to fit the model is:

```
eq index1: index1
eq index2: index2
eq factoreq: index1 index2
gllamm y index1 index2 gender1 gender2, nocons i(id school) /*
        */ nrf(1 2) eqs(factoreq index1 index2) ip(m) nip(7) adapt
```

Note that the equations `index1` and `index2` used for the random effects at the school level include a one variable each, while the equation `factoreq` used for the factor at the student level includes two variables.

The commands to recover the variance-covariance parameters at the student level are:

```
nlcom var1: [id1_1]index1^2+exp(2*[lns1]_cons)
nlcom var2: [id1_1]index1^2*[id1_1l]index2^2+exp(2*[lns1]_cons)
nlcom cov: [id1_1l]index2*([id1_1]index1)^2
nlcom rho: ([id1_1l]index2*[id1_1]index1^2)/sqrt(([id1_1]index1^2+/*
          */exp(2*[lns1]_cons))*([id1_1]index1^2*[id1_1l]index2^2+/*
           */exp(2*[lns1]_cons)))
```

The results are reported in the last column of Table 7.

3.3 Multivariate binary-Normal response model

To illustrate how to fit a mixed response model and make a comparison with the review on MLwiN, we dichotomized the first response and fitted a single-level (dropping the school level) null model.

The total score of written papers y_1 was recoded as $y_1^{\text{bin}}=1$ if $y_1>80$ and 0 otherwise (preserving missing values). The fitted model is a single-level bivariate binary-Normal null model:

$$\begin{cases} \tilde{y}_{1j} = \alpha_1 + \varepsilon_{1j}, & \left\{y_{1j}^{\text{bin}} = 1\right\} \Leftrightarrow \left\{\tilde{y}_{1j} > 0\right\} \\ y_{2j} = \alpha_2 + \varepsilon_{2j} \end{cases} \qquad (13)$$

$$\Omega_\varepsilon = \begin{pmatrix} \sigma_{\varepsilon_1}^2 & \\ \sigma_{\varepsilon_{12}} & \sigma_{\varepsilon_2}^2 \end{pmatrix}$$

Since $\tilde{y}_{1j}$ is unobservable, its variance is not estimable and in fact it is implicitly fixed by choosing the link (e.g. $\sigma_{\varepsilon_1}^2 = \pi^2/3$ for the logit link). There are two estimable variance-covariance: $\sigma_{\varepsilon_2}^2$ and $\sigma_{\varepsilon_{12}}$.

To speed up the computation, the model can be reparametrized using a single random effect with factor loadings, with the loading on the first equation fixed to one:

$$\begin{cases} \tilde{y}_{1j} = \alpha_1 + u_j + e_{1j}, & \left\{y_{1j}^{\text{bin}} = 1\right\} \Leftrightarrow \left\{\tilde{y}_{1j} > 0\right\} \\ y_{2j} = \alpha_2 + \lambda u_j + e_{2j} \end{cases} \qquad (14)$$

$$\Omega_e = \begin{pmatrix} \theta_1^2 & \\ 0 & \theta_2^2 \end{pmatrix} \qquad u_j \sim N(0, \varphi^2)$$

Note that the residuals in (14) are decomposed as $\varepsilon_{1j}=u_j+e_{1j}$ and $\varepsilon_{2j}=\lambda u_j+e_{2j}$. Therefore, $\text{var}(\varepsilon_1) = \varphi^2 + \theta_1^2$, $\text{var}(\varepsilon_2) = \lambda^2\varphi^2 + \theta_2^2$, $\text{cov}(\varepsilon_1, \varepsilon_2) = \lambda\varphi^2$.

However, there are four variance-covariance parameters in (14) (φ^2,θ_1^2, θ_2^2 and λ), while there are only two estimable quantities: the residual variance of y_2, namely $\lambda^2\varphi^2 + \theta_2^2$, and the correlation between the total residuals of the two equations, namely $\text{corr}(\varepsilon_{1j}, \varepsilon_{2j}) = \lambda\varphi^2 / \sqrt{(\varphi^2 + \theta_1^2)(\lambda^2\varphi^2 + \theta_2^2)}$. Therefore, it is necessary to impose two restrictions. One restriction directly stems from the binary nature of the first equation, so θ_1^2 is implicitly fixed to a value determined by the link function (here we use the logit link, so $\theta_1^2=\pi^2/3$). The second restriction needed for identification must be stated explicitly: here we fix the factor variance to one, i.e. $\varphi^2=1$ (for a discussion of drawbacks and alternative constraints, see Skrondal and Rabe-Hesketh, 2004, pp. 107-108).

The gllamm syntax for such a model is:

```
eq factoreq: index1 index2
constraint define 2 [id1_1]index1=1
gllamm ymixed index1 index2, nocons i(id) nrf(1) eqs(factoreq) /*
```

```
        */constraint(2) lv(index) link(logit identity) /*
        */ fv(index) family(binomial gaussian) nip(10) adapt
```

The `constraint` command fixes to one the standard deviation of the factor. The `lv` option declares the equations to which the links listed in `link()` apply. The `fv` option has the same role w.r.t. `family()`.

The results are reported in the last column of Table 8. Since the standard deviation of ε_1 is $\sqrt{1+\pi^2/3}$ instead of $\sqrt{\pi^2/3}$, the α_1 estimate needs to be rescaled by a factor 1.142 to get the estimate for (approximately) a logit model. In this case $\hat{\alpha}_1 = -6.093$, so the rescaled value is -5.336.

3.4 A Model for Meta-analysis

Meta analysis pools results from different studies estimating the same quantity of interest. The main focus is on the between-study variation.

The data can be available in different forms: (a) individual level data for each study, and (b) an "effect-size" estimate for each study along with its standard error.

In case (a) an ordinary two-level model with study level random effects and covariates at the individual and study levels can be fitted.

In case (b) the "effect-sizes" are regressed on some covariates at the study level given the standard deviations of the studies. Let y_j denotes the "effect-size" estimate for study j, and s_j its known standard-error. The model is:

$$y_j = \alpha + \beta x_j + u_j + e_j \qquad (15)$$

$$u_j \sim N(0, \sigma_u^2), \qquad e_j \sim N\left(0, s_j^2\right),$$

where σ_u^2 is the variance parameter to be estimated for the between-study variability and x_j is an explanatory variable at the study level.

To let e_j have the known standard deviation s_j, one can use the `s()` option, together with the `constraint()` option, to specify a model for the log-standard deviation.

To illustrate the procedure, we use the example from the meta-analysis of teacher expectancy effects (Raudenbush and Bryk, 2002). The effect-size and standard deviation, together with an explanatory variable for 19 clusters, are available. The syntax for fitting model (15) with `gllamm` is as follows:

```
generate lns = ln(sd2)/2
eq het: lns
constraint define 1 [lns1]lns=1
gllamm effect week, i(studyID) constraint(1) s(het) nip(15) adapt
```

The results, with and without the covariate week, are reported in Table 9.

4. Final remarks

The gllamm command of Stata is a very powerful tool for fitting a wide range of multilevel models. In fact, the command is designed to fit the models belonging to the broad GLLAMM class, whose theory is developed in Skrondal and Rabe-Hesketh (2004). The close connection between the estimation command and a methodological framework turns out to be a great virtue, especially when the researcher wishes to extend the basic models.

In this review we considered only the basic random effects models. Other relevant models handled by gllamm include: survival models, sample selection models, ordinal models with thresholds depending on covariates, factor and item response models (IRT), structural equation models (SEMs), latent class models and measurement error models. All these models can be defined with an arbitrary number of nested levels.

Even if there is no point-and-click interface, the gllamm syntax is intuitive and easy to learn. However, the gllamm user has to know certain features of Stata, e.g. how to save the estimates or how to define a constraint.

To obtain diagnostics, such as residual plots or influence measures, one can use the gllapred post-estimation command to compute many kind of residuals and other diagnostic measures and then use Stata to produce custom-made plots.

The estimation method of gllamm is full information ML via numerical quadrature, assuming Gaussian random effects. Is also possible to assume a discrete distribution for the random effects leading to non-parametric ML. Other estimation methods, such as REML and MCMC, are not implemented.

The main limit of gllamm lies in the computational efficiency. In fact, even if the options can always be set to obtain accurate estimates, when the number of random effects (latent variables) is over four or five the computational time is likely to become excessive for many users. Since one of the factors influencing the computational time is the number of records in the data set, it is crucial to collapse the data set and use the weight option whenever possible. Moreover, good starting values can speed up estimation: e.g., when adding quadrature points or adding or dropping covariates it is useful to use the previous estimates as starting values via the from() option.

Despite its extraordinary flexibility, there are certain models that gllamm cannot directly fit: cross-classified models, multiple membership models, models with spatial/temporal autoregressive structures, models with "mixed" random effects (in the sense that some random effects are continuous and some are discrete). Nonetheless, some of these models can be fitted with gllamm using a trick, e.g. the cross-classified model can be fitted by defining a pseudo top-level unit (Rabe-Hesketh and Skrondal, 2005, ch. 8).

References

Langford, I.H., Bentham, G. and McDonald, A. (1998) Multilevel modelling of geographically aggregated health data: a case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* **17**: 41-58.

Lindsay, B.G. (1995) Mixture Models: Theory, Geometry and Applications. NSF-CBMS Regional Conference Series in Probability and Statistics, vol. 5. Hayward, CA: Institute of Mathematical Statistics.

Huq, N.M. and Cleland, J., (1990) Bangladesh Fertility Survey 1989, Main Report. Dhaka: National Institute of Population Research and Training (NIPORT).

McGrath, K. and Waterton, J. (1986) British Social Attitudes, 1983-1986 Panel Survey, Technical Report. London, Social and Community Planning research.

Rabe-Hesketh, S., Pickles, A. and Skrondal, A. (2003). Correcting for covariate measurement error in logistic regression using nonparametric maximum likelihood estimation. *Statistical Modelling* **3** (3), 215-232.

Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2002). Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal* **2** (1), 1-21.

Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2005). Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* **128** (2), 301-323.

Rabe-Hesketh, S. and Skrondal, A. (2005). Multilevel and Longitudinal Modeling using Stata. College Station, TX: Stata Press.

Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2004). Generalized multilevel structural equation modelling. *Psychometrika* **69** (2), 167-190.

Rabe-Hesketh, S. and Skrondal, A. (2006). Multilevel Modeling of Complex Survey Data. *Journal of the Royal Statistical Society, Series A.* In press.

Rabe-Hesketh, S., Skrondal, A., and Pickles, A. (2004) *GLLAMM Manual. U.C. Berkeley Division of Biostatistics Working Paper Series.* Working Paper 160.
http://www.bepress.com/ucbbiostat/paper160

Rasbash, J., Steele, F., Browne, W., Prosser, B. (2005). *A user's guide to MLwiN version 2.0.*, University of Bristol.

Raudenbush, S.W. and Bryk, A., (2002) *Hierarchical Linear Models. Applications and Data Analysis Methods*. Newbury Park, CA: Sage.

Skrondal, A. and Rabe-Hesketh, S. (2004). Generalized latent variable modeling. Boca Raton: Chapman & Hall/CRC.

Table 1 - Two-level Normal model A (4059 students nested within 65 schools)

<table>
<tr><th>Estimation method</th><th>Syntax specification to fit the model</th><th>-2logL</th><th>$\hat{\beta}_0$</th><th>$\hat{\beta}_1$</th><th>$\hat{\beta}_2$</th><th>$\hat{\beta}_3$</th><th>$\hat{\beta}_4$</th><th>$\hat{\sigma}^2_{u0}$</th><th>$\hat{\sigma}^2_{e0}$</th><th>Time conv.</th></tr>
<tr><td><em>REML</em></td><td>xi: xtmixed exam standlrt gender i.schgend || schid: , variance</td><td>9347.6</td><td>-.0094
(.0779)</td><td>.5598
(.0124)</td><td>.1674
(.0341)</td><td>-.1590
(.0894)</td><td>.0187
(.1261)</td><td>.0858
(.0178)</td><td>.5625
(.0126)</td><td>&lt;1’’</td></tr>
<tr><td><em>ML</em></td><td>xi: xtmixed exam standlrt gender i.schgend || schid:, ml variance</td><td>9325.4</td><td>-.0092
(.0763)</td><td>.5600
(.0124)</td><td>.1672
(.0341)</td><td>-.1590
(.0873)</td><td>.0187
(.1232)</td><td>.0811
(.0165)</td><td>.5623
(.0126)</td><td>&lt;1’’</td></tr>
<tr><td><em>ML adapt. quadr.8pt</em></td><td>xi: gllamm exam standlrt gender i.schgend, i(schid) adapt nip(8)</td><td>9325.4</td><td>-.0092
(.0763)</td><td>.5600
(.0125)</td><td>.1672
(.0341)</td><td>-.1590
(.0873)</td><td>.0187
(.1232)</td><td>.0811
(.0165)</td><td>.5623
(.0126)</td><td>3’12’’</td></tr>
<tr><td rowspan="2"><em>NPML (5 mass points)</em></td><td rowspan="2">xi: gllamm exam standlrt gender i.schgend, i(schid) ip(f) nip(5) from(b) gateaux(-2, 2, 20) lf0(`k' `ll')</td><td rowspan="2">9312.6</td><td>-.0255
(.0576)</td><td>.5603
(.0123)</td><td>.1706
(.0339)</td><td>-.1270
(.0437)</td><td>.0003
(.0645)</td><td>.0822*</td><td>.5616
(.0125)</td><td rowspan="2">Total
≈5’
Last step
≈1’</td></tr>
<tr><td colspan="7">Locations of random effects: -0.598, -0.321, -0.137, 0.100, 0.524
Probabilities of random effects: 0.071, 0.096, 0.289, 0.408, 0.137</td></tr>
</table>

Note: before estimation the command char schgend[omit] 3 was submitted to define the 3rd category of schgend as the reference category.
Model A: Variance component model with covariates standlrt, gender and schgend and Gaussian random effects.

Fixed parameters: β_0 (constant), β_1 (standlrt), β_2 (gender), β_3 (schgend 1), β_4 (schgend 2);

Random parameters: σ^2_{u0} (level 2), σ^2_{e0} (level 1).

NPML: discrete distribution of the random effects. The matrix b contains the estimates from the previous step, while the local variables k and ll contain the number of parameters and the log-likelihood from the previous step.
* This value is obtained from the locations and probabilities of the random effects shown below.

Table 2 - Two-level Normal models B and C (4059 students nested within 65 schools) – `xtmixed` REML

Model building	Fixed and random parameters.	Syntax specification to fit the model	REML Estimates (SE)	Time to convergence
B: Variance component with interaction, `gender*standlrt`	*Fixed* ($\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$) *Random* Level 2: ($\sigma^2_{u_0}$) Level 1: ($\sigma^2_{e_0}$)	`xi: xtmixed exam i.gender*standlrt i.schgend || schid: , variance`	$\hat{\beta}_0$ =-0.0094 (0.0779) $\hat{\beta}_1$ =0.5626 (0.01837) $\hat{\beta}_2$ =0.1673 (0.0341) $\hat{\beta}_3$ =-0.1588 (0.0894) $\hat{\beta}_4$ =0.0189 (0.1261) $\hat{\beta}_5$ =0.0051 (0.0246) $\hat{\sigma}^2_{u_0}$ =0.0858 (0.0178) $\hat{\sigma}^2_{e_0}$ =0.5627 (0.0126) -2LogL=9353.20	<1''
C: Random slopes of `standlrt` effect	*Fixed* ($\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5$) *Random* Level 2: ($\sigma^2_{u_0}, \sigma_{u01}, \sigma^2_{u_1}$) Level 1: ($\sigma^2_{e_0}$)	`xi: xtmixed exam i.gender*standlrt i.schgend || schid:standlrt, cov(unstruct) variance`	$\hat{\beta}_0$ = -0.01197 (0.0742) $\hat{\beta}_1$ = 0.5503 (0.0257) $\hat{\beta}_2$ = 0.1686 (0.0338) $\hat{\beta}_3$ = -0.1779 (0.0821) $\hat{\beta}_4$ = -0.0004 (0.1162) $\hat{\beta}_5$ = 0.0069 (0.0295) $\hat{\sigma}^2_{u_0}$ = 0.0837 (0.0174) σ_{u01} = 0.0205 (0.0070) $\sigma^2_{u_1}$ = 0.0153 (0.0047) $\hat{\sigma}^2_{e_0}$ = 0.5504 (0.0124) -2logL= 9308.24	<2''

Note: before estimation the command `char schgend[omit] 3` was submitted to define the 3rd category of `schgend` as the reference category.

Table 3 - Two-level Normal models D and E (4059 students nested within 65 schools) – `gllamm` ML spherical adaptive quadrature with 7 points

Model building	Fixed and random parameters.	Syntax specification to fit the model	ML Estimates (SE)	Time to convergence

D: Level 1 variance heterogeneity by 'girl' (1=boy, 0=girl)	*Fixed*
	$(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$
	Random
	Level 2: $(\sigma^2_{u_0}, \sigma_{u_{01}}, \sigma^2_{u_1})$
	Level 1: $(\sigma^2_{e_1}, \sigma^2_{e_2})$
		`eq sch_c: cons`
		`eq sch_lrt: standlrt`
		
		`gen boy=gender==0`
		`gen girl=gender==1`
		
		`eq vargen: boy girl`
		
		`xi: gllamm exam i.gender*standlrt i.schgend, i(schid) nrf(2) eqs(sch_c sch_lrt) s(vargen) ip(m) adapt nip(7)`
			$\hat{\beta}_0$ =-0.0114 (0.0728)
			$\hat{\beta}_1$ =0.5506 (0.0261)
			$\hat{\beta}_2$ =0.1685 (0.0340)
			$\hat{\beta}_3$ =-0.1787 (0.0801)
			$\hat{\beta}_4$ =-0.0008 (0.1141)
			$\hat{\beta}_5$ =0.0070 (0.0297)
			$\hat{\sigma}^2_{u_0}$ =0.0795 (0.0164)
			$\sigma_{u_{01}}$ =0.0203 (0.0068)
			$\sigma^2_{u_1}$ =0.0149 (0.0046)
			$\hat{\sigma}^2_{e_1}$ =0.5876 (0.0209)
			$\sigma^2_{e_2}$ =0.5252 (0.0153)
			-2logL= 9275.07
				≈12’

E: Level 1 variance as an exponential function of 'standlrt'	*Fixed*
	$(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)$
	Random
	Level 2: $(\sigma^2_{u_0}, \sigma_{u_{01}}, \sigma^2_{u_1})$
	Level 1: (σ^2_e, δ)
		`eq varlrt: cons standlrt`
		
		`xi: gllamm exam i.gender*standlrt i.schgend , i(schid) nrf(2)  eqs(sch_c sch_lrt) s(varlrt) ip(m) adapt nip(7)`
			$\hat{\beta}_0$ =-0.0110 (0.0732)
			$\hat{\beta}_1$ =0.5519 (0.0256)
			$\hat{\beta}_2$ =0.1683 (0.0338)
			$\hat{\beta}_3$ =-0.1786 (0.0806)
			$\hat{\beta}_4$ =-0.0044 (0.1142)
			$\hat{\beta}_5$ =0.0074 (0.0295)
			$\hat{\sigma}^2_{u_0}$ =0.0798 (0.0164)
			$\sigma_{u_{01}}$ =0.0207 (0.0067)
			$\sigma^2_{u_1}$ =0.0144 (0.0045)
			$\hat{\sigma}^2_e$ =0.5496 (0.0124)
			δ =-0.0541 (0.0231)
			-2logL= 9275.60
				≈12’

Note: before estimation the command `char schgend[omit] 3` was submitted to define the 3rd category of `schgend` as the reference category.

Table 4 - `gllamm` specifications for 2-level binary response models

Random intercept model (ML 8pt adaptive quadrature)		
parameters	*logit link*	*probit link*
constant	-1.6902 (0.1477)	-1.0286 (0.087)
urban	0.7324 (0.1195)	0.4491 (0.0727)
age	-0.0266 (0.0079)	-0.0163 (0.0048)
nchild=1	1.1093 (0.158)	0.6702 (0.0947)
nchild=2	1.3765 (0.1748)	0.8348 (0.1049)
nchild≥3	1.3456 (0.1796)	0.8148 (0.1074)
$\sigma_{u_0}^2$	0.2155 (.0733)	0.07984 (.0269)
-2logL	2413.34	2412.74
Time to convergence	51"	1’1"
Random slope model (ML spherical 7pt adaptive quadrature)		
	logit link	*probit link*
Constant	-1.7128 (0.1605)	-1.0419(0.0949)
Urban	0.8162 (0.1714)	0.5003 (0.1043)
age	-0.0265 (0.0080)	-0.0164 (0.0049)
nchild=1	1.1260 (0.1603)	0.6816 (0.0959)
nchild=2	1.3682 (0.1772)	0.8306 (0.1062)
nchild≥3	1.3555 (0.1828)	0.8245 (0.1092)
$\sigma_{u_0}^2$	0.3906 (0.1293)	0.1436 (.04688)
$\sigma_{u_{01}}$	-0.4064 (.1753)	-0.1504 (.0643)
$\sigma_{u_1}^2$	0.6665 (0.3212)	0.2468 (.1187)
-2logL	2398.36	2397.56
Time to convergence	2’ 19”	2’ 44”

Table 5 - Poisson models for malignant melanoma mortality data. ML adaptive quadrature.

	Two-level model		*Three-level model*	
	No overdispersion	*With overdispersion*	*No overdispersion*	*With overdispersion*
Fixed effects				
α	-0.1386 (0.0494)	-0.1472 (0.0482)	-0.0640 (0.1336)	-0.0865 (0.1299)
Uvb	-0.0344 (0.0100)	-.0393 (0.0099)	-0.0282 (0.0114)	-0.0335 (0.0114)
Random effects				
Region σ_u^2	0.1699 (0.0310)	0.1556 (0.0295)	0.0484 (0.0109)	0.0406 (0.0105)
Nation σ_v^2			0.1372 (0.0723)	0.1289 (0.0682)
ψ^2		0.0148 (0.0051)		0.0147 (0.0051)
-2logL	2250.30	2232.53	2190.62	2173.35
Overdispersion LRT		17.77		17.27
Time to converge	6”	1’ 27”	3’	30’ 24”
Quadrature points	8	8	10	10

Table 6 - Estimates of two-level ordinal and nominal models for Social Attitudes data

Fixed effects	Proportional odds model	*Fixed effects*	Multinomial model
$\alpha^{(1)}$	-4.1418 (0.6323)	$\beta_0^{(1)}$	-1.9176 (0.2729)
$\alpha^{(2)}$	-2.3981 (0.5991)	$\beta_0^{(2)}$	-1.2622 (0.2380)
$\alpha^{(3)}$	0.4034 (.5894)	$\beta_0^{(3)}$	0.2999 (0.2036)
$\alpha^{(4)}$	1.3404 (0.5907)	$\beta_0^{(4)}$	-0.3693 (0.2128)
$\alpha^{(5)}$	2.2307 (0.5933)	$\beta_0^{(5)}$	-0.3932 (0.2133)
$\alpha^{(6)}$	3.2732 (0.5921)	$\beta_0^{(6)}$	-0.2718 (0.2111)
Religion=2	1.9593 (0.6343)		
Religion=3	0.6907 (0.6906)		
Religion=4	2.7778 (0.6502)		
Random effects		*Random effect*	
σ_u^2	5.1330 (0.6976)	σ_u^2	5.5416 (1.0732)
-2logL	3171.29		3426.0046
Quadrature points	8		8
Time to converge	1’40”		2’ 18”

Table 7 - Bivariate Normal model of student scores. Students nested in schools. Adaptive spherical quadrature.

	Model with fixed level one variance		*Model with a factor at student level*
Fixed effects	*7 pt*	*15 pt*	*7 pt*
α_1	49.002 (1.1993)	49.008 (0.9321)	49.008 (0.9308)
α_2	69.680 (0.6265)	69.623 (1.1725)	69.623 (1.1723)
β_1	-2.442 (0.5745)	-2.493 (0.5607)	-2.493 (0.5607)
β_2	6.745 (0.5681)	6.757 (0.6708)	6.757 (0.6708)
Random effects: *Student level*			
$\sigma^2_{\varepsilon_1}$	$\varphi^2+\theta_1^2$=125.405 (4.2488)	$\varphi^2+\theta_1^2$=124.433 (4.3363)	$\varphi^2+\theta^2$=124.432 (4.3354)
$\sigma^2_{\varepsilon_2}$	$\varphi^2+\theta_2^2$=175.634 (5.4168)	$\varphi^2+\theta_2^2$=180.070 (6.2499)	$\lambda^2\varphi^2+\theta^2$=180.070 (6.250)
$\sigma_{\varepsilon_1\varepsilon_2}$	40.638 (2.4855)	72.749 (4.1521)	$\lambda\varphi^2$=72.748 (4.1519)
corr (ε_1,ε_2)	0.274	0.486	0.486
School level			
$\sigma^2_{v_1}$	48.738 (10.4885)	46.565 (9.3531)	46.586 (9.2283)
$\sigma^2_{v_2}$	71.210 (7.0668)	75.194 (14.6729)	75.197 (14.6539)
$\sigma_{v_1v_2}$	4.830 (3.7582)	24.937 (8.9916)	24.944 (8.9713)
corr(v_1,v_2)	0.082	0.421	0.421
Parameters actually outputted by gllamm			
θ^2	fixed at 20.086	fixed at 20.086	74.365 (2.7247)
var(u$_1$)	105.320 (4.2488)	104.348 (4.3363)	
var(u$_2$)	155.549 (5.4168)	159.984 (6.2499)	
Cov(u$_1$,u$_2$)	40.638 (2.4855)	72.749 (4.1521)	
φ^2			50.067 (4.115)
λ			1.453 (0.0654)
-2logL	26956.19	26799.51	26799.51
Time to converge	$\approx 2^h30'$	$\approx 43^h$	39′

Table 8 - Bivariate models of student scores. Schools ignored. 10 pt adaptive quadrature.

	Responses	
Fixed effects	*Normal-Normal*	*Binary (logit)-Normal*
α_1	46.370 (0.3197)	-5.336* (0.4112)
α_2	73.318 (0.3882)	73.398 (0.3929)
Random effects: *Student level*		
$\sigma^2_{\varepsilon_1}$	$\varphi^2+\theta^2=$ 178.210 (6.0788)	fixed by the link
$\sigma_{\varepsilon_1\varepsilon_2}$	101.977 (5.8553)	
$\sigma^2_{\varepsilon_2}$	$\lambda^2\varphi^2+\theta^2=$ 265.417 (8.9960)	$\lambda^2+\theta^2=$266.261 (9.0783)
Corr(ε_1,ε_2)	0.4689 (0.0195)	0.4558 (0.0205)
Parameters actually outputted by gllamm		
θ^2	110.905 (3.9533)	91.555 (49.5759)
φ^2	67.3045 (5.694)	constrained to 1
λ	1.515 (0.0699)	13.218 (1.9006)
-2logL	27804.44	14604.12
Time to converge	2′17″	≈ 3′

* This value is obtained by rescaling the estimate -6.093 by a factor 1.142 to adjust the scale of the latent response (see Section 3.3, last sentence)

Table 9 Model estimates based on 19 effect-sizes data in meta analysis. 15 pt adaptive quadrature.

Parameters	*Null model*	*Model with covariate*
β_0	0.0787 (0.0521)	0.4086 (0.0871)
β_1		-0.1580 (0.0359)
σ^2_u	0.0131 (0.0201)	0.0000 (0.0000)
-2logL	-3.1620	6.1321
Time to converge	5″	5″