GLLAMM Manual

Technical Report 2001/01

Department of Biostatistics and Computing

Institute of Psychiatry, King's College, University of London

Sophia Rabe-Hesketh
Department of Biostatistics and Computing
Institute of Psychiatry
King's College, University of London

Andrew Pickles
School of Epidemiology and Health Science and CCSR
The University of Manchester

Anders Skrondal
Department of Epidemiology
National Institute of Public Health, Oslo

October 28, 2001

Introduction and Disclaimer

`gllamm` is a Stata program to fit GLLAMMs (Generalised Linear Latent and Mixed Models).

GLLAMMs are 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 (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 generalised linear models or generalised linear mixed models, multilevel factor or latent trait models, ordered latent class models and multilevel structural equation models.

Chapter 1 of this manual describes the models, Chapter 2 describes the program and subsequent chapters give examples. The examples are arranged in chapters according to the structure of the models (chapters 2 to 5) and, for complex response processes, according to the type of response (chapters 6 to 9). There are obvious gaps in these chapters and we hope to include more examples soon.

Potential users of `gllamm` are reminded to be extremely careful if using this program for serious statistical analysis. It is the user's responsibility to check that the models are identified, that the program has converged, that the quadrature approximation used is adequate, etc. The manual provides quite a number of examples of different model structures where `gllamm` yields results identical to those reported elsewhere obtained using more specialized programs. Though this provides some validation of the code, nonetheless some bugs may remain. Both the program and the manual will continually be updated.

Bug reports, comments and suggestions are all welcome! Please contact Sophia Rabe-Hesketh at spaksrh@iop.kcl.ac.uk. We would also be grateful if you could let us know of any publications (including 'in press') using `gllamm`.

The program and this manual can be downloaded from

http://www.iop.kcl.ac.uk/iop/departments/biocomp/programs/gllamm.html

where the data used in the examples of this manual are also available. An older (out of date) version of the program (`gllamm6`) is available via the Stata Technical Bulletin (Rabe-Hesketh, Pickles and Taylor, 2000).

We would like to acknowledge Colin Taylor for his help in the early stages of `gllamm` development, in particular his guidance on recursive programming and quadrature.

Contents

Chapter 1

Generalised linear latent and mixed models

GLLAMMs (Generalised Linear Latent And Mixed Models) are 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. Examples of models in this class are multilevel generalised linear models or generalised linear mixed models, multilevel factor or latent trait models, ordered latent class models and multilevel structural equation models.

GLLAMMs can be defined by specifying:

1. The conditional expectation of the responses given the latent and observed explanatory 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.

These specifications are covered in Sections 1.1 to 1.4.

Terminology: latent variables and levels

The models include latent or unobserved variables represented by the elements of a vector $\mathbf{u}$. As we will see later, *the latent variables can be interpretable as random effects (random intercepts and coefficients) or factors and we will use these terms interchangeably.*

In addition, the latent variables can vary at different 'levels' so that we can have level 2 factors, level 3 random effects, etc. In the case of hierarchical data, the term 'level' is often used to describe the position of a unit of observation within a hierarchy of units, typically reflecting the sampling design. Here level 1 units are nested in level 2 units which are nested in level 3 units, a typical example being pupils in classes in schools. In this context, a random effect is said to vary at a given level, e.g. at the school level, if it varies between schools but, for a given school, is constant for all classes and pupils belonging to that school.

We will use the term level to refer to the position of a 'unit' within the structure of a model, not necessarily within the sampling structure of the data. The models assume that *lower level*

'units' are conditionally independent given the higher level latent variables and the explanatory variables. An example where the distinction between the levels of a hierarchical data structure and the levels of the model becomes important are multivariate multilevel models. Here the variables of the multivariate response are often treated as level 1 units so that the levels of the model do not correspond with the levels reflecting the hierarchical structure of the data (except, possibly in the case of repeated measures or longitudinal data).

1.1 Conditional expectation of the responses

We refer to a particular response simply as y, omitting subscripts for the units of observation. The conditional expectation of the response y given two sets of explanatory variables $\mathbf{x}$ and $\mathbf{z}$ and the vector of latent variables $\mathbf{u}$ is specified via a link function $g()$ and a linear predictor η as

$$g(E[y|\mathbf{x},\mathbf{u},\mathbf{z}]) = \eta \tag{1.1}$$

where the link can be any of the links used in generalised linear mixed models. For a model with L levels, and M_l latent variables at level l, the linear predictor has the form

$$\eta = \boldsymbol{\beta}'\mathbf{x} + \sum_{l=2}^{L}\sum_{m=1}^{M_l} u_m^{(l)}\boldsymbol{\lambda}_m^{(l)\prime}\mathbf{z}_m^{(l)} \tag{1.2}$$

with the first element of $\boldsymbol{\lambda}_m^{(l)}$ set to 1, i.e.

$$\lambda_{m1}^{(l)} = 1. \tag{1.3}$$

The elements of $\mathbf{x}$ are explanatory variables associated with the 'fixed' effects $\boldsymbol{\beta}$, $u_m^{(l)}$ is the mth latent variable at level l and $\mathbf{u}$ in equation (1.1) is the vector of latent variables,

$$\mathbf{u} = (u_1^{(2)}, u_2^{(2)}, \cdots u_{M_2}^{(2)}, u_1^{(3)}, u_2^{(3)}, \cdots u_{M_3}^{(2)}, \cdots, u_1^{(L)}, u_2^{(L)}, \cdots u_{M_L}^{(L)}) \tag{1.4}$$

Each latent variable is multiplied by a linear combination of explanatory variables $\boldsymbol{\lambda}_m^{(l)\prime}\mathbf{z}_m^{(l)}$. Here the superscript of $\mathbf{z}_m^{(l)}$ denotes that the corresponding latent variable varies at level l (generally, $\mathbf{z}_m^{(l)}$ will vary at a lower level than l). The vector $\mathbf{z}$ in (1.1) has the same structure as $\mathbf{u}$. The latent variables at the same level are generally mutually correlated whereas latent variables at different levels are independent.

The model in equation (1.2) includes multilevel generalised linear models (or generalised linear mixed models) and multilevel factor models as special cases.

1.1.1 Multilevel generalised linear models

There are a large number of books on multilevel models, see for example Bryk and Raudenbush (1992) and Kreft and De Leeuw (1998) for texts on linear multilevel models and Snijders and Bosker (1999), Longford (1993), Goldstein (1995) and McCulloch and Searle (2001) for texts on both linear and generalised linear multilevel models.

To write down a multilevel generalised linear model, or generalised linear mixed model, simply use one explanatory variable $z_{m1}^{(l)}$ for each latent variable (with $\lambda_{m1}^{(l)} = 1$) so that the latent variable can be interpreted as a random coefficient or random slope. Multilevel generalised linear models are therefore a special case of GLLAMMs with

$$\eta = \boldsymbol{\beta}'\mathbf{x} + \sum_{l=2}^{L}\sum_{m=1}^{M_l} u_m^{(l)} z_{m1}^{(l)} \tag{1.5}$$

where, typically, $z_{11}^{(l)} = 1$ so that there is a random intercept at each level. Omitting the random terms yields a generalised linear model.

An example of a three level model (using subscripts i, j, k for levels 3,2,1, respectively) with random intercepts at levels 2 and 3 and a random coefficient at level 2 is

$$\eta_{ijk} = \boldsymbol{\beta}'\mathbf{x}_{ijk} + u_{1ij}^{(2)} + u_{2ij}^{(2)} z_{21}^{(2)} + u_{1i}^{(3)}. \tag{1.6}$$

(Here $z_{11}^{(2)}$ and $z_{11}^{(3)}$ were set to 1.)

Multilevel generalised linear models are sometimes defined by first writing down the relationship between the response variable and the level 1 covariates, where some coefficients vary at level 2. Models for these coefficients are then specified where the coefficients are regressed on level 2 covariates and have level 2 residuals, etc. For a two-level linear model (using subscripts i and j for levels 3 and 2), the relationship between the response variable and the level 1 covariates has the form

$$y_{ij} = b_{0i} + b_{1i}x_{1ij} + \epsilon_{ij}, \tag{1.7}$$

where some of the parameters (here the intercept and slope) vary between level 2 units as indicated by the i subscript. Regressions of these parameters on level 2 covariates are then specified with residual error terms at level 2, e.g.,

$$\begin{aligned} b_{0i} &= \gamma_{00} + \gamma_{01}z_{1i} + u_{0i} \\ b_{1i} &= \gamma_{10} + \gamma_{11}z_{1i} + u_{1i}. \end{aligned} \tag{1.8}$$

Again, some parameters may vary between level 3 units (here they do not) in which case regressions for these parameters are specified. This method of model specification is used, for example, in Bryk and Raudenbush (1992) and in the HLM program (Bryk *et al.*,1996).

When a multilevel model has been specified in the way described above, we can substitute the models for the coefficients into the model for the observed responses to obtain the *reduced form*, a single model equation having the form of equation (1.5). Substituting the expressions for b_{0i} and b_{1i} into the first equation, we obtain:

$$\begin{aligned} y_{ij} &= \gamma_{00} + \gamma_{01}z_{1i} + u_{0i} + (\gamma_{10} + \gamma_{11}z_{1i} + u_{1i})x_{1ii} + \epsilon_{ij} \\ &= \gamma_{00} + \gamma_{01}z_{1i} + \gamma_{10}x_{1ij} + \gamma_{11}z_{1i}x_{1ij} + u_{0i} + u_{1i}x_{1ij} + \epsilon_{ij} \end{aligned} \tag{1.9}$$

By forming a new variable $x_{2ij} = z_{1i}x_{1ij}$, it is easy to see that this equation is a special case of equation (1.5). (Note that the level 1 error term ϵ_{ij} does not appear in (1.5) because it is not part of the linear predictor in the generalised linear model formulation.)

We have only used one explanatory variable per latent variable to set up these standard models. However, the use of several explanatory variable enables us to allow the random effects variances to vary between groups of individuals. For example, consider a two level random intercept model. If the level 1 units j are the measurement occasions of a longitudinal study and the level 2 units i are children, we can allow the intercept variance to differ between boys and girls by defining dummy variables for boys and girls, say z_{bij} and z_{gij}, respectively, and specifing

$$\eta_{ij} = \boldsymbol{\beta}'\mathbf{x} + u_i^{(2)}(z_{bij} + \lambda_g z_{gij}) \tag{1.10}$$

so that the random intercept variance is $\text{var}(u_i^{(2)})$ for boys and $\lambda_g^2\text{var}(u_i^{(2)})$ for girls. Heteroscedasticity of the random effects can be specified at any of the levels. In addition, we can allow the level 1 variances to differ (see Section 1.2).

Examples of multilevel generalised linear models are given in Chapter 3. Papers using `gllamm` for multilevel generalised linear models include Rabe-Hesketh *et al.* (2001c) and Leese *et al.* (2001) for applications in psychiatry and Dohoo *et al.* (2001) and Stryhn *et al.* (2000) for applications in veterinary medicine.

1.1.2 Multilevel factor models

For a treatment of factor models for normal and non-normal responses, see Bartholomew and Knott (1999). Multilevel factor models for continuous data are discussed in Longford (1993).

By treating the variables of a multivariate response as level 1 units in a multilevel dataset (the original units become level 2 units), and by defining appropriate dummy variables, factor models can be defined. As mentioned earlier, we will use the term 'level' to refer to the position of a latent variable in the structure of the model; for a non-hierarchical multivariate dataset consisting of responses to questionnaire items by subjects, the common factor will be a level 2 latent variable. To see this, consider a simple example. Let there be up to J variables $j = 1, \cdots, J$ observed on each individual i and stack the variables into a single response vector indexed ij. This is shown in the table below:

subject i	variable j	z_{11}	z_{12}	$\cdots$	y
1	1	1	0	$\cdots$	y_{11}
1	2	0	1	$\cdots$	y_{12}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2	1	1	0	$\cdots$	y_{21}
2	2	0	1	$\cdots$	y_{22}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

A single level unidimensional factor model can be written as

$$\begin{aligned}\eta_{ij} &= \beta_1 x_{1ij} + \beta_2 x_{2ij} + \cdots + u_{1i}^{(2)}(\lambda_{11}^{(2)} z_{11ij}^{(2)} + \lambda_{12}^{(2)} z_{12ij}^{(2)} + \lambda_{13}^{(2)} z_{13ij}^{(2)} + \cdots) \\ &= \beta_j + u_{1i}^{(2)}\lambda_{1j}^{(2)} \end{aligned} \tag{1.11}$$

where $\lambda_{11}^{(2)} = 1$ and $x_{pij} = z_{1pij}^{(2)}$ with

$$z_{1pij}^{(2)} = \begin{cases} 1 & \text{if p = j} \\ 0 & \text{otherwise} \end{cases} \tag{1.12}$$

In equation (1.11), β_j is the intercept, $u_{1i}^{(2)}$ is the common factor and $\lambda_{1j}^{(2)}$ is the factor loading for the jth variable. The scale of the factor is identified through the constraint that the first factor loading equals 1. For normally distributed responses, the specific factors are simply the level 1 error terms ϵ_{ij} since $y_{ij} = \eta_{ij} + \epsilon_{ij}$. In factor models for non-normal responses, the level 1 variability is implicit in the distribution family of the chosen generalised linear model (see Section 1.2).

A two factor model at a single level can be defined as

$$\eta_{ij} = \beta_1 x_{1ij} + \beta_2 x_{2ij} + \cdots + u_{1i}^{(2)}(\lambda_{11}^{(2)} z_{11ij}^{(2)} + \lambda_{12}^{(2)} z_{12ij}^{(2)} + \cdots) + u_{2i}^{(2)}(\lambda_{21}^{(2)} z_{21ij}^{(2)} + \lambda_{22}^{(2)} z_{22ij}^{(2)} + \cdots) \tag{1.13}$$

Here, some of the variables would typically load on factor 1, $u_{1i}^{(2)}$ and the others on factor 2, $u_{2i}^{(2)}$. (The dummy variables $z_{1pij}^{(2)}$ and $z_{2pij}^{(2)}$ must be defined appropriately). Certain restrictions need to be imposed on the factor loadings and/or the covariance matrix of the factors for these higher dimensional factor models to be identified.

A two level factor model (unidimensional at levels 1 and 2) can be defined as

$$\begin{aligned}\eta_{ijk} &= \beta_1 x_{1ijk} + \beta_2 x_{2ijk} + \cdots + u_{1ij}^{(2)}(\lambda_1 z_{11ijk}^{(2)} + \lambda_2^{(2)} z_{12ijk}^{(2)} + \cdots) \\ &\quad + u_{(J+1)i}^{(3)}(\lambda_1 z_{11ijk}^{(3)} + \lambda_2^{(3)} z_{12ijk}^{(2)} + \cdots) + u_{1i}^{(3)} z_{11ijk}^{(2)} + u_{2i}^{(3)} z_{12ijk}^{(2)} + \cdots \end{aligned} \tag{1.14}$$

where $z_{1pij}^{(2)} = z_{1pij}^{(3)}$ are dummy variables for the items as in equation (1.12), $u_{1ij}^{(2)}$ is the lower level common factor, $u_{(J+1)i}^{(3)}$ is the higher level common factor and $u_{1i}^{(3)}$, $u_{2i}^{(3)}$, etc. are specific factors at the higher level. The level 3 latent variables are assumed to be mutually independent.

Although this method of defining factor models through the use of dummy variables is not very elegant, a great advantage of the specification is that missing values on any of the variables are allowed and pose no extra problem in the estimation. Since estimation is by maximum likelihood, the parameter estimates are consistent if the data are missing at random (MAR), see Little and Rubin (1987). In addition, unlike the usual model specification for structural equation models, this setup allows unbalanced longitudinal data to be modelled where individuals are measured at different sets of time points. In addition, hybrid models, containing both factors and random coefficients at several levels can easily be defined using this setup.

Examples of factor models are given in Chapter 4. Rabe-Hesketh and Skrondal (2001) discuss the use of factor models (estimated in `gllamm`) for structuring the covariance matrix of multivariate categorical responses.

1.2 Conditional distributions of the responses

The conditional distribution of the responses given the explanatory variables and random effects is specified via a family and a link function (see McCullagh and Nelder, 1989). Currently available are the following links and families:

Links
identity
reciprocal
logarithm
logit
probit
scaled probit
complimentary log-log

Families
Gaussian
gamma
Poisson
binomial

For ordered and unordered categorical responses, the following models are available:

Polytomous responses
ordinal logit
ordinal probit
ordinal compl. log-log
scaled ordinal probit
multinomial logit

Offsets can be included in the linear predictor and linear constraints applied to any of the parameters.

For the Gaussian and gamma distributions as well as the scaled probit link, the variance parameter can be allowed to differ between groups of observations or to depend on explanatory variables, therefore allowing for level 1 heteroscedasticity.

The available links and families allow many different response processes to be modelled including continuous responses, dichotomous or ordinal responses, counts, continuous or discrete time (interval censored) to event (survival) data and first choice and ranking data.

Different links and families can be combined for different responses in order to model responses of mixed type. This allows many different types of problems to be modelled, for example logistic regression with measurement errors in a continuous covariate. Mixing of the identity link and Gauss family with the probit link and binomial family allows censored normally distributed responses to be modelled (as in tobit), e.g. when there are ceiling and/or floor effects.

Examples of models using particular links and families can be found using the Index. The analysis of continuous survival times is discussed in Chapter 7 and the analysis of nominal responses and rankings is discussed in Chapter 9. Examples of models with mixed responses are discussed in Chapter 6.

Leese *et al.* (2001) use `gllamm` to specify heteroscedasticity at levels 1 and 2. Rabe-Hesketh and Pickles (1999; 2001) mix the identity and logit links to estimate logistic regression models with covariate measurement error. Rabe-Hesketh *et al.* (2001d) analyse different types of discrete time survival models and Skrondal and Rabe-Hesketh (2001) analyse nominal data and rankings.

1.3 Structural equations for the latent variables

Books on structural equation models include Dunn, Everitt and Pickles (1993) and Bollen (1989). The models discussed so far can be viewed as a measurement models specifying the relationship between observed responses (or indicators) and latent and observed explanatory variables. In addition, we can specify relationships among the latent variables as well as regressions of latent variables on explanatory variables. These relationships can be written as a matrix equation for the vector of latent variables $\mathbf{u}$ whose M elements are the latent variables varying at levels 2 to L,

$$\mathbf{u} = (u_1^{(2)}, u_2^{(2)}, \cdots u_{M_2}^{(2)}, u_1^{(3)}, u_2^{(3)}, \cdots u_{M_3}^{(3)}, \cdots, u_1^{(L)}, u_2^{(L)}, \cdots u_{M_L}^{(L)}). \tag{1.15}$$

(Remember that the latent variables can be interpretable as variance components, factors or random coefficients.) The 'structural' equation for the latent variables has the form

$$\mathbf{u} = \mathbf{B}\mathbf{u} + \mathbf{\Gamma}\mathbf{w} + \boldsymbol{\zeta} \tag{1.16}$$

where $\mathbf{B}$ is an $M \times M$ upper diagonal matrix, $\mathbf{w}$ is a vector of q covariates, $\mathbf{\Gamma}$ is an $M \times q$ design matrix and $\boldsymbol{\zeta}$ is a vector of M errors of disturbances where each element of $\boldsymbol{\zeta}$ varies at the same level as the corresponding element of $\mathbf{u}$.

The regressions of latent variables on other latent variables must be such that the elements of the $\mathbf{u}$ vector within a given level can be permuted in such a way to make the $\mathbf{B}$ matrix upper diagonal. This implies that there are no simultaneous effects with latent variable 1 regressed on latent variable 2 and vice versa. The expression for the Mth element of $\mathbf{u}$ can be substituted into the expression for $M-1$th element which can be substituted into the expression for $M-2$nd element, etc. (i.e., the relationship is recursive). When these expressions for the latent variables are substituted into equation (1.2), we obtain an equation of the same form as equation (1.2) with constraints among the parameters.

Since the lower level latent variables come before the higher level ones in the $\mathbf{u}$ vector, an upper diagonal $\mathbf{B}$ matrix ensures that lower level latent variables can be regressed on higher but not the reverse since it would not make sense to regress a higher level latent variable on a lower level one.

1.3.1 MIMIC models

MIMIC (Multiple Indicator Multiple Causes) models are factor models (where observed variables are indicators of the factor) combined with regressions of factors on explanatory variables (the explanatory variables are the causes). Here, there are no relationships among the latent variables and instead of equation (1.16), we only require simply the equation

$$\mathbf{u} = \mathbf{\Gamma w} + \boldsymbol{\zeta}. \tag{1.17}$$

An example of a MIMIC model is described in Section 6.1.

1.3.2 General multilevel structural equation models

Combining (1.16) with (1.2) and making use of the ability to model responses of mixed types allows a very large range of latent variable models to be defined, see Rabe-Hesketh *et al.* (2001a). For example, a level 2 random coefficient in a survival model may be regressed on a level 3 factor whose (level 3) indicators are dichotomous. A chapter on these models will be added soon.

1.4 Distribution of the latent variables

The structure of the latent variables is specified by the number of levels L (and the variables defining these levels, e.g. pupil, school etc.) and the number of latent variables M_l at each level. Latent variables at the same level are assumed to be correlated with each other, unless the user specifies zero correlations. The latent variables at different levels are assumed to be independent (except if a lower level latent variable is explicitly regressed on a higher level one). The interpretation of the latent variable as a factor or random coefficient depends on the form of the linear predictor in (1.2).

If a latent variable is regressed on (another) latent or observed variable, we need to specify the distribution of the disturbances $\boldsymbol{\zeta}$; otherwise we specify the distribution of $\mathbf{u}$ directly. The latent variables at a level l may be assumed to have a

- multivariate normal distribution with zero mean and covariance matrix $\mathbf{\Sigma}_l$ at level l
- discrete distribution, having non-zero probability on a finite number of points (of dimensionality equal to the number of latent variables, M_l, at level l)

The discrete distribution can be interpreted as representing a number of latent classes which are homogeneous in the unobserved characteristics 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) is achieved (Lindsay *et al.*, 1991). The Gateaux derivative method can be used to determine the number of masses required for the NPML solution (see Heckman and Singer (1984), Follmann and Lambert (1989) and Davies and Pickles (1987))). Starting with a small number of masspoints, say two, the likelihood is maximized. A further point is introduced if a location can be found at which introduction of a very small new mass increases the likelihood when all other parameters are held constant at their previous maximum likelihood values. If such a location can be found, a new point is introduced and the likelihood

maximized. The starting values are the parameters estimates of the previous model with a new mass at the location yielding the greates an increase in log-likelihood. This procedure is repeated until no location can be found at which introduction of a small mass increases the likelihood.

Most examples in the manual assume multivariate normally distributed latent variables except for the examples in Chapter 5 and Section 9.4. Papers using `gllamm` with discrete random effects include Maughan *et al.* (2000) on latent trajectory models, Rabe-Hesketh and Pickles (2001) and Rabe-Hesketh *et al.* (2001c) on nonparametric maximum likelihood estimation and Rabe-Hesketh and Pickles (1999) on a latent transition model.

Chapter 2

The gllamm program

The gllamm program runs within Stata 6 or 7 (StataCorp. 2001) using a similar syntax to Stata's own estimation commands. After estimating a model using gllamm, gllapred can be used to obtain the posterior means and standard deviations of the latent variables.

2.1 Implementation

gllamm uses Stata's ml with method d0 to maximise the likelihood. (See the Stata reference manuals under ml and maximize; for details of the modified Newton Raphson algorithm, see also Gould and Sribney (1999)). For a 2-level model, the likelihood is given by

$$\prod_i \int \{\prod_j f(y_{ij}|\mathbf{x}_i, \mathbf{u}_i)\} g(\mathbf{u}_i) d\mathbf{u}_i \tag{2.1}$$

where $f(y_{ij}|\mathbf{x}_i, \mathbf{u}_i)$ is the conditional density of the response variable given the latent and explanatory variables and $g(\mathbf{u}_i)$ is the prior density of the latent variables. When the latent variables are discrete, the integral becomes a sum of the form

$$\prod_i \sum_r \pi_r \prod_j f(y_{ij}|\mathbf{x}_i, \mathbf{u}_i = \mathbf{z}_r) \tag{2.2}$$

where the locations $\mathbf{z}_r$ and masses π_r are freely estimated. For a single normally distributed latent variable, the same expression is used to approximate the likelihood, where locations and masses are given by Gaussian quadrature.

If there are $M_2 > 1$ correlated (multivariate normal) latent variables at level 2, we express them as a linear combination of uncorrelated random effects $\mathbf{v}$, $\mathbf{u} = \mathbf{L}\mathbf{v}$ where $\mathbf{L}$ is a lower triangular matrix. The multiple integral is then approximated by summing over $\mathbf{v} = \mathbf{z}_r$ using M_2 nested sums. The elements of $\mathbf{L}$ are estimated by gllamm and the covariance matrix of the random effects is given by $\mathbf{L}'\mathbf{L}$ so that $\mathbf{L}$ is simply the Cholesky decomposition of the covariance matrix.

Ordinary Gaussian quadrature sometimes performs poorly because there are insufficient locations $\mathbf{z}_r$ under the peak of the integrand in (2.1). Adaptive quadrature (Naylor and Smith, 1982; Liu and Pierce, 1994) can be used to scale and shift the locations for each latent variable v_m according to the spread τ_m and location μ_m of the integrand with respect to v_m. The integrand is proportional to the joint posterior density of the latent variables. Therefore appropriate scale and location parameters are the posterior standard deviations and means of $\mathbf{v}$. Since these quantities depend on the parameter estimates, gllamm iterates between updating the parameter estimates for given τ_m and μ_m and updating the τ_m and μ_m for given parameter estimates. See Rabe-Hesketh *et al.*(2001) for details.

In (2.1), the contribution to the likelihood from the ith level 2 unit is found by integrating the product of the contributions from the level 1 units inside the level 2 unit over the level 2 random effects distribution. In a three level model, the contribution from a 3-level unit is found by integrating the product of contributions from the level 2 units inside the level 3 unit over the level 3 random effects distribution. The likelihood for an n-level model is therefore computed using a recursive algorithm.

The parameters are transformed to ensure that they lie within their permitted ranges. For the normal (or gamma) density, the log of the standard deviation (or coefficient of variation) at level 1 is estimated to ensure a positive estimate on the natural scale. When quadrature is used, the Cholesky decomposition of the covariance matrix of the random effects at each level is estimated to ensure a semi positive definite covariance matrix. When there are no correlations, this corresponds to estimating the standard deviations directly where the sign of these estimates is arbitrary. When R discrete mass-points are specified for the random effects at a level, $R-1$ log odds are estimated to give the R probabilities and $R-1$ locations are estimated directly for each random effect. The last location is determined by constraining the mean of the discrete distribution to zero. The variance of the random effects distribution is not estimated directly but follows from the locations and masses. The variances are estimated as

$$\mathrm{var}(u_m) = \sum_r \hat{\pi}_r \hat{z}_{rm}^2 \tag{2.3}$$

and the covariances are estimated as

$$\mathrm{cov}(u_m, u_{m'}) = \sum_r \hat{\pi}_r \hat{z}_{rm} \hat{z}_{rm'} \tag{2.4}$$

where $\hat{\pi}_r$ and $\hat{z}_m$ are the estimated probabilities and locations. Instead of centering the location of the discrete distribution around the mean, we can also estimate R locations freely and remove the corresponding fixed effects from the linear predictor. In the equations above, we must then replace $\hat{z}_{rm}$ by $\hat{z}_{rm} - \sum_r \hat{\pi}_r \hat{z}_{rm}$.

Approximate standard errors for the back-transformed parameter estimates are obtained using the delta-method (except for the variances and covariances of the discrete random effects distributions). Note that the standard error of variance estimates should not be used to construct Wald tests because the null value is on the border of the parameter space. Likelihood ratio test should be used instead, keeping in mind that the alternative hypothesis and p-value are one sided.

Linear constraints can be specified for the transformed versions of the parameters that are used during maximisation. For example, two parameters can be set equal, a parameter can be constrained to a particular value or one parameter can be specified to be twice as large as another.

The posterior means and standard deviations of the latent variables can be estimated for both discrete and continuous latent variables using `gllapred`. If there is a single random effect in a two level model, the posterior mean is estimated as

$$\hat{u}_i = \sum_r \hat{z}_r \hat{w}_r \tag{2.5}$$

where $\hat{w}_r$ is the estimated posterior probability that the latent variable equals $\hat{z}_r$ given by

$$\hat{w}_r = \frac{\hat{\pi}_r \prod_j f(y_{ij}|\mathbf{x}_{ij}, u_i = \hat{z}_r)}{\sum_r \hat{\pi}_r \prod_j f(y_{ij}|\mathbf{x}_{ij}, u_i = \hat{z}_r)} \tag{2.6}$$

2.2 Installing gllamm

The version of the program described in this manual is available at

http://www.iop.kcl.ac.uk/IoP/Departments/BioComp/programs/gllamm.html

it can also be found by searching for the program `gllamm` at the Boston IDEAS archive maintained by Kit Baum at

http://ideas.uqam.ca/ideas/search.html

An earlier verion (January 2000) of `gllamm` was published in the Stata Technical Bulletin (STB 53, Sg 129) but this version is now out of date. A later version (June 2001) is described in Rabe-Hesketh *et al.* (2001b) and (2001e).

The easiest way of installing the current version of the program is to use the net commands from within Stata:

```
net from http://www.iop.kcl.ac.uk/IoP/Departments/BioComp/programs
net describe gllamm
net install gllamm
net get gllamm
```

Where the last command is optional; it downloads the auxiliary files. Alternatively, `net install` from the IDEAS archive using:

```
net from http://fmwww.bc.edu/RePEc/bocode/
net cd g
net describe gllamm
net install gllamm
```

Another possibility is to store the files *gllamm.ado*, *gllam_ll.ado*, *remcor6.ado*, *gllamm.hlp*, *gllapred.ado gllapred.hlp* (downloadable from the above web-sites) in the directory where personal ado-files are stored (e.g. `c:\ado\stbplus`) or to store them in any other directory and issue the following command before using `gllamm`:

`adopath +` *dirname*

Once `gllamm` has been installed, help is available as for all of Stata's own commands.

2.3 Running gllamm

`gllamm` runs in Stata 6 and 7 (StataCorp, 2001). Anyone planning to use `gllamm` 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 using `gllamm`, see the Appendix for a brief introduction to Stata. Sections 2.3.1 and 2.3.2 describe the syntax of `gllamm` and `gllapred` following the style of the Stata Reference Manuals.

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 maximise 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.

An easy way to speed up the program is to collapse the data as much as possible and use frequency weights (see for example Section 3.2.1). If there are several identical level 2 units, level 2 weights can be used. It is also a good idea to start with fewer integration points to obtain some initial estimates and then pass these estimates as starting values to `gllamm`, increasing the number of quadrature points (see for example Section 7.2.2). This way, the accuracy of the quadrature approximation can be assessed (see also Section 2.3.3). It is also recommended to first estimate the simplest model of interest (e.g. using very few predictors) and then introduce additional features, again passing the parameter estimates from the simpler model to `gllamm` as starting values for the more complicated model.

The program does not check whether a model is identified. Using the `trace` option to monitor convergence may help identify problems since warning messages that the log-likelihood is nonconcave may appear shortly before apparent convergence (if such messages appear in the beginning, this is no cause for concern). `gllamm` prints out the condition number, defined as the square root of the ratio of the largest to smallest eigenvalues of the Hessian matrix. From our experience so far, large condition numbers do not necessarily imply poor identification; however, it is unlikely that a low condition number is obtained when the model is not identified.

2.3.1 Syntax for `gllamm`

The full syntax with all its options looks overwhelming because a single program can estimate such a wide range of different models. For most models, the command would be no longer than a single line and the syntax closely follows that of similar Stata commands. The reader may find it easier to read Chapter 3 as an introduction to `gllamm`. This section on the syntax could be used as a reference. To find examples in the manual of the use of particular options, also see the Index.

The data need to be in 'long' form with all responses stacked into a single variable, see the Appendix, Stata's `reshape` command and the 'data preparation' sections in this manual. The full syntax with all available options is

`gllamm` [*varlist*] [`if` *exp*] [`in` *range*] , `i`(*varlist*) [`nrf`(#,...,#) `eqs`(*eqnames*)

`nocorrel` `noconstant` `offset`(*varname*) `weight`(*varname*) `family`(*families*)

`fv`(*varname*) `denom`(*varname*) `link`(*links*) `lv`(*varname*) `expanded`(*varname varname* string) `basecategory(#)` `s`(*varname*) `thresh`(*eqnames*) `ip`(*string*) `nip`(#,...,#)

`adapt` `geqs`(*eqnames*) `bmatrix`(*matname*) `constraints`(#,...,#) `from`(*matrix*)

`long` `lf0`(# #) `gateaux`(###) `search`(#) `maximize_options` `nodifficult`

level(#) eform allc trace nolog noest eval init dots]

where [*varlist*] gives the response variable followed by the explanatory variables associated with the fixed effects. The families and links are:

families	links
gaussian	identity
poisson	log
gamma	reciprocal
binomial	logit
	probit
	cll (complimentary log-log)
	ologit (o stands for ordinal)
	oprobit
	ocll
	mlogit
	sprobit (scaled probit)
	soprobit (scaled ordinal probit)

Options

Structure of the model

i(*varlist*) gives the variables that define the hierarchical, nested clusters, from the lowest level (finest clusters) to the highest level, e.g., `i(pupil class school)`.

nrf(#,...,#) specifies the number of latent variables at each level of clustering, i.e. for each variable in i(*varlist*). The default is `nrf(1,...,1)`.

eqs(*eqnames*) specifies the equation names (defined before running `gllamm`) for the linear predictors multiplying the latent variables. The equations for the level 2 random effects a listed first, followed by those for the level 3 random effects, etc., the number of equations per level being specified in the `nrf()` option. If required, constants should be explicitly included in the equation definitions using variables equal to 1. If the option is not used, the latent variables are assumed to be random intercepts and only one random effect is allowed per level. The first lambda coefficient is set to one. The other coefficients are estimated together with the (co)variance(s) of the random effect(s)

nocorrel may be used to constrain all correlations to zero if there are several random effects at any of the levels and if these are modelled as multivariate normal.

geqs(*eqnames*) specifies regressions of latent variables on explanatory variables. The second character of the equation name indicates which latent variable is regressed on the variables used in the equation definition, e.g. `f1:a b` means that the first latent variable is regressed on `a` and `b` (without a constant).

bmatrix(*matrix*) specifies a square matrix **B** of regression coefficients for the dependence of the latent variables on other latent variables. The matrix must be upper diagonal and have number of rows and columns equal to the total number of random effects. This option only makes sense together with the `nocorrel` option.

`constraint`(*clist*) specifies the constraint numbers of the linear constraints to be applied. Constraints are defined using the `constraint` command; see `help constraint`. To find out the equation names needed to specify the constraints, run `gllamm` with the `noest` and `trace` options.

`noconstant` omits the constant term from the fixed effects equation.

`offset`(*varname*) specifies a variable to be added to the linear predictor with corresponding regression coefficient fixed at 1 (e.g. log exposure time for Poisson regression).

`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 equal to one.) For example, if the level 1 units are occasions (or panel waves) in longitudinal data and the level 2 units are individuals, and the only variable used in the analysis is a binary variable `result`, we can collapse dataset A into dataset B by defining level 1 weights as follows:

A

ind	occ	result
1	1	0
1	2	0
2	3	0
2	4	1
3	5	0
3	6	1

B

ind	occpat	result	wt1
1	1	0	2
2	2	0	1
2	3	1	1
3	4	0	1
3	5	1	1

The two occasions for individual 1 in dataset A have the same result. The first row in B therefore represents two occasions (occasions 1 and 2) as indicated by `wt1`. The variable `occpat` labels the unique patterns of responses at level 1.

The two individuals 2 and 3 in dataset B have the same pattern of results over the measurement occasions (both have two occasions with values 0 and 1). We can therefore collapse the data into dataset C by using level 2 weights:

B

ind	occpat	result	wt1
1	1	0	2
2	2	0	1
2	3	1	1
3	4	0	1
3	5	1	1

C

indpat	occpat	result	wt1	wt2
1	1	0	2	1
2	2	0	1	2
2	3	1	1	2

The variable `indpat` labels the unique patterns of responses at level 2 and `wt2` indicates that `indpat` 1 in dataset C represents one individual and `indpat` 2 represents two individuals, i.e., all the data for individual 2 are replicated once. Collapsing the data in this way can make `gllamm` run considerably faster.

Densities, links, latent variable distribution and quadrature method

`family`(*families*) specifies the family (or families) to be used for the conditional densities. The default is `gaussian`. Also available are `binomial`, `poisson` and `gamma`. Several families may be given in which case the variable allocating families to observations must be given using `fv`(*varname*).

`fv`(*varname*) is required if mixed responses requiring more than a single family of conditional distributions are being analyzed. The variable indicates which family applies to which observation. A value of one refers to the first family specified in `family()`, etc.

`denom`(*varname*) gives the variable containing the binomial denominator for the responses whose family was specified as binomial. The default denominator is 1.

`link`(*links*) specifies the links to be used for the conditional densities (`identity, logit, probit, log, reciprocal, cll, ologit`, etc.). If a single family is specified, the default link is the canonical link. Several links may be given in which case the variable allocating links to observations must be given using `lv`(*varname*). Feasible choices of link depend upon the distributions of the covariates and the choice of conditional error and random effects distributions.

`lv`(*varname*) is the variable whose values indicate which link applies to which observation.

`expanded`(*varname varname string*) is used together with the mlogit link and specifies that the data have been expanded as illustrated below:

```
A                    B
choice               response altern selected
   1                     1       1       1
   2                     1       2       0
                         1       3       0
                         2       1       0
                         2       2       1
                         2       3       0
```

where the variable `choice` is the multinomial response (possible values 1,2,3), the `response` labels the original lines of data, `altern` gives the possible responses or 'alternatives' and `selected` is an indicator for the response that was given. The syntax would be `expanded(response selected m)` and `altern` would be used as the dependent variable. This expanded form allows the user to have alternative specific covariates, apply different random effects to different alternatives and have different alternative sets for different individuals. The third argument is `o` if one set of coefficients should be estimated for the explanatory variables and `m` if one set of coefficients is to be estimated for each category of the response escept the reference category.

`basecategory`(#) When the mlogit link is used, this specifies the value of the response to be used as the reference category. This option is ignored if the `expanded()` option is used with the third argument equal to `m`.

`s`(*eqname*) specifies that the log of the standard deviation (or of the coefficient of variation) at level 1 for normally (or gamma) distributed responses is modelled by the linear predictor defined by `eqname`. This is necessary if the variance is heteroscedastic. For example, if dummy variables for groups are used in the definition of *eqname*, different variances are estimated for different groups.

`thresh`(*eqnames*) specifies equation(s) for the thresholds for ordinal response(s). One equation is specified for each ordinal response. The purpose of this option is to allow the effects of some covariates to be different for different categories of the ordinal outcome rather than assumming a constant effect - the proportional odds assumption if the ologit link is used. Variables used in the model for the thresholds cannot appear in the fixed part of the linear predictor.

`ip`(*string*) if string is `g`, Gaussian quadrature points are used and if string is `f`, the mass-points are freely estimated. The default is Gaussian quadrature. With the `ip(f)` option, the `nip-1` mass-point locations are estimated the last being determined by setting the mean of the mass-point distribution to 0. The `ip(fn)` option can be specified to estimate `nip` masses freely - the user must make sure that the mean is not modelled in the linear predictor, e.g. by specifying the `nocons` option.

`nip`(#,...,#) specifies the number of integration points or masses to be used for each integral or summation. When quadrature is used, a value may be given for each random effect. When freely estimated masses are used, a value may be given for each level of the model. If only one argument is given, the same number of integration points will be used for each summation. The default value is 8.

`adapt` causes adaptive quadrature to be used instead of ordinary quadrature. This option cannot be used with the `ip(f)` or `ip(f0)` options.

Starting values

`from`(*matrix*) specifies the matrix (one row) to be used for the initial values. Note that the column-names and equation-names have to be correct (see `help matrix`), unless the `copy` option is used. The matrix may be obtained from a previous estimation command using e(b). This is useful if another explanatory variable needs to be added or the number of masses needs to be increased. (The `skip` option must be used of variables are dropped.)

`long` may be used with the `from(matrix)` option when parameter constraints are used to indicate that the matrix of initial values corresponds to the unconstrained model, i.e. it has more elements than will be estimated.

`lf0`(# #) gives the number of parameters and the log-likelihood for a likelihood ratio test to compare the model to be estimated with a simpler model. A likelihood ratio chi-squared test is only performed if the `lf0()` option is used.

`gateaux`(# # #) may be used with method `ip(f)` or `ip(fn)` options to increase the number of mass-points by one from a previous solution with parameter estimates specified using `from(matrix)`. The number of parameters and log-likelihood of the previous solution must be specified using the `lf0(# #)` option. 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. (If there are several random effects, this search is done in each dimension resulting in a regular grid of search points.) 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 non-parametric maximum likelihood estimate.

`search`(#) causes the program to search for initial values for the random effects at level 2 (in range 0 to 3). The argument specifies the number of random searches. This option may only be used with `ip`(g) and when `from`($matrix$) is not used.

Estimation options and output

`maximize_options` see `ml`. The most useful one is `iterate(#)` since by default the program does not limit the number of iterations. If the matrix of initial parameter estimates specified using the `from(matrix)` option, the `skip` option causes redundant parameters to be skipped (ignored). The `copy` option can be used if the matrix of starting values does not have the correct equation and/or column names.

`nodifficult` causes ml not to use the difficult option, see [R] maximise.

`level`(#) specifies the confidence level in percent for confidence intervals of the fixed coefficients.

`eform` causes the exponentiated estimates and confidence intervals of the fixed coefficients to be displayed.

`allc` causes all estimated parameters to be displayed in a regression table (including the raw random effects parameters) in addition to the usual output.

`trace` is one of the maximize_options, see [R] maximize. In addition to displaying the details of the maximum-likelihood iterations, it displays details of the model being fitted.

`nolog` suppresses output for maximum likelihood iterations.

`noest` is used to prevent the program from carrying out the estimation. This may be used with the trace option to check that the model is correct and get the information needed to set up a matrix of initial values. Global macros are available that are normally deleted. Particularly useful may be M_initf and M_initr, matrices for the parameters (fixed part and random part, respectively).

`eval` causes the program to simply evaluate the log likelihood for values passed to `gllamm` using the `from(matrix)` option.

`init` causes the program to compute initial estimates of fixed effects only

`dots` causes a dot to be printed (if used together with trace) every time the likelihood evaluation program is called by ml. This helps to assess how long gllamm is likely to take to run and reassures the user that it is making some progress when it is very slow.

2.3.2 Syntax for `gllapred`

After estimating the parameters of a model using `gllamm`, we can run the 'post-estimation' command `gllapred` to obtain predictions or estimates of the latent variables both in the continuous and discrete case. Both posterior means and standard deviations of the latent variables are provided. In multilevel regression models, the posterior means are also referred to as empirical Bayes predictions, shrinkage estimates, or higher level residuals. In factor models, the posterior means are regression factor scores. The posterior standard deviations can be interpreted as standard errors of the posterior means although they do not take into account the fact that the parameters are estimated.

`gllapred` *varname* [, `xb` `u` `p` `s` `linpred` `adapt`]

where *varname* is the 'prefix' used for the variables that will contain the predictions.

Options

`xb` The fixed effects part of the linear predictor is returned in *varname* including the offset.

`u` The posterior means and standard deviations of the latent variables are returned in *varname*`m1`, *varname*`m2`, etc. and *varname*`s1`, *varname*`s2`, etc., respectively, where the order of the latent variables is the same as in the call to `gllamm`. In the case of quadrature, the number of quadrature points used is also the same as in the previous call to `gllamm`.

`p` can only be used for two-level models estimated using the ip(f) option. gllapred returns the posterior probabilities in *varname*1, *varname*2, etc., giving the probabilities of classes 1,2, etc. gllapred also prints out the (prior) probability and location matrices to help interpret the posterior probabilities.

`s` returns the scale. This is useful if the s() otion was used in gllamm to specify level 1 heteroscedasticity.

`linpred` returns the linear predictor including the fixed and random effects part where posterior means are substituted for the latent variables or random effects in the random part.

`adapt` if the gllamm command did not use the adapt option, gllapred will use ordinary quadrature for computing the posterior means and standard deviations unless the adapt option is used in the gllapred command.

2.3.3 Some comments on quadrature

Bock (1985) suggested that in most applications 10 quadrature are sufficient in one dimension, 5 per dimension in two and 3 per dimension in three or more dimensions. However, Bartholomew (1987) points out investigations performed by Shea (1984) indicating that at least 20 points may be necessary to achieve reasonable accuracy. Similarly, in logistic regression with covariate measurement error problems, Crouch and Spiegelman (1990) suggest that 20 quadrature points or more are often needed to adequately approximate the marginal log-likelihood. Skrondal (1996) carried out a simulation for a two factor model with 3 ordinal items loading on each factor finding substantial bias in the parameter estimates when only 5 quadrature points were used per dimension whereas 20 quadrature points performed adequately. In practice, the adequacy of the number of quadrature points used in a particular application should be checked. A good discussion of potential numerical problems with quadrature and suggestions for discovering these problems can be found in the Stata Reference Manual (2001) under `quadchk`. There it is pointed out that the method works better for small (level 2) cluster sizes or if the intraclass correlation is not too great. Intuitively, for a simple two level random intercept model, this can be understood by imagining the joint conditional distribution of the level 1 responses, $\prod_j f(y_{ij}|\mathbf{x}_i, u_i)$ of a given cluster i given the value of the random intercept u_i, plotted against the value of the random intercept. If the 'true' realisation of the random intercept for that cluster is large, (i.e. the responses are all substantially 'higher' than the overall mean) the graph will have a sharp peak in the tail of the prior distribution $g(u_i)$ of the random intercept. If we do not use sufficient quadrature points, there may not be enough quadrature locations under the peak of the integrand, $g(u_i)\prod_j f(y_{ij}|\mathbf{x}_i, u_i)$. In fact, since increasing

the number or quadrature points increases mostly the range of locations rather than their density, it may not be possible to adequately approximate the marginal likelihood using quadrature.

Lesaffre and Spiessens (2001) discuss this problem in relation to binary data with high intraclass correlation and Albert and Follmann (2000) for Poisson responses. In the Poisson case, if the responses (usually counts) are high, numerical problems with quadrature can occur also for smaller clusters because the conditional distribution of a single level 1 unit can have a very sharp peak. A method that is likely to work better than ordinary quadrature is adaptive quadrature (Liu and Pierce, 1994) implemented in SAS PROC NLMIXED. This method has just been implemented in `gllamm`, see Rabe-Hesketh *et al.* (2001e).

Obviously, problems with quadrature are likely with normally distributed responses. We would therefore not recommend using ordinary quadrature. Adaptive quadrature may give good estimates but it may be better to use software that does not use any approximations for normally distributed responses (e.g. Mplus by Muthén and Muthén (1998), S-Plus lme, MLwiN (Goldstein, 1998), etc.)

A number of programs for generalised linear mixed models, including MLwiN and HLM use MQL/PQL (Breslow and Clayton, 1993). Taylor expansions are used to linearise the relationship between response and linear predictor; both first and second order expansions are available in MLwiN. The PQL/MQL methods work best if the level 2 clusters are very large; in the case of smaller clusters, the variances of the random effects tend to be underestimated (Lin and Breslow, 1995, Breslow and Lin, 1995, Rodriguez and Goldman, 1995).

Since quadrature tends to work better for smaller cluster sizes, PQL may work well when quadrature does not and vice versa. Rabe-Hesketh *et al.* (2001c) Dohoo *et al.* (2001) and Stryhn *et al.* (2000) compare quadrature with PQL for particular examples using simulations. In Section 9.3.3 of this manual, we compare MQL/PQL with quadrature in the case of nominal (unordered categorical) responses.

2.3.4 Some comments on nonparametric maximum likelihood estimation

The following papers evaluate the performance of nonparametric maximum likelihood (NPML): Davies (1987), Hu *et al.* (1998), Magder and Zeger (1996), Follmann and Lambert (1989), and Rabe-Hesketh and Pickles (2001).

Chapter 3

Multilevel generalised linear models

3.1 Three level random intercept logistic regression

Three groups of subjects, a group of patients with schizophrenia, their first degree relatives and an independent control group, completed a neuropsychological test called the Tower of London. In this computerised task, subjects are given a starting arrangement of 3 disks on 3 rods and are asked to move the disks among the rods to achieve a 'target' arrangement, if possible, in a specified minimum number of moves. The level of difficulty is increased by increasing the minimum number of moves required. We will analyse the binary response, whether the task was completed using the minimum number of moves, for three levels of difficulty. Taking into account the nesting of subjects in families, this becomes a three level problem. (The data are also analysed in Rabe-Hesketh *et al.* (2001c).)

We will use indices i, j and k for families, subjects and measurement occasions, respectively. The binary responses y_{ijk} may be modelled by a generalised linear mixed model with linear predictor

$$\eta_{ijk} = \beta_0 + \beta_1 x_{Rijk} + \beta_2 x_{Sijk} + u_{ij}^{(2)} + u_i^{(3)} \tag{3.1}$$

where x_{Rijk} and x_{Sijk} are dummy variable for the relatives and patients with schizophrenia, respectively, $u_{ij}^{(2)}$ is the random intercept for subject j in family i and $u_i^{(3)}$ is the random effect for family i. The random effects are assumed to be independently normally distributed.

3.1.1 Data preparation

The dataset is called *towerl.dta*. The responses are stacked into a variable called `dtlm` (dichotomised tower of london moves). The variables `id`, `famnum` and `group` are the subject, family and group identifiers respectively and `level` codes the levels of difficulty. A listing of these variables for observations 19 to 27 is given below.

```
. list id famnum group level dtlm in 19/27

           id     famnum      group      level       dtlm
 19.        7          6          1          0          0
 20.        7          6          1          1          0
 21.        7          6          1         -1          1
 22.        8         15          1          0          0
 23.        8         15          1          1          0
 24.        8         15          1         -1          1
 25.        9         10          1         -1          0
 26.        9         10          1          0          0
 27.        9         10          1          1          0
```

3.1.2 Model fitting

A two-level model with a random effect for subjects could be fitted using Stata's `xtlogit` command with the following syntax:

```
xi: xtlogit dtlm i.group level, i(id) quad(20)
```

The syntax for the same model using `gllamm` is

```
xi: gllamm dtlm i.group level, i(id) family(binom) link(logit) nip(20)
```

The syntax is therefore very similar to that of `xtlogit` except that the logit link and binomial family are specified as in Stata's `glm` command and the `nip()` option specifies the number or quadrature points ('integration points') to be used.

The following output is obtained:

```
. xi: gllamm dtlm i.group level, i(id) family(binom) link(logit) nip(20)
i.group           _Igroup_1-3         (naturally coded; _Igroup_1 omitted)

number of level 1 units = 677
number of level 2 units = 226

Condition Number = 4.4745542

gllamm model

log likelihood = -305.95929

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Igroup_2 |  -.1690688    .334248    -0.51   0.613    -.8241828    .4860452
   _Igroup_3 |  -1.022688   .3938465    -2.60   0.009    -1.794613   -.2507631
       level |  -1.648835   .1933503    -8.53   0.000    -2.027795   -1.269875
       _cons |  -1.482656   .2835593    -5.23   0.000    -2.038422   -.9268896
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): 1.6752845 (.66195499)
------------------------------------------------------------------------------
```

The number of level 1 units (here measurement occasions) and the number of level 2 units (here individuals) are listed first, followed by the condition number and the log-likelihood. (The condition number will be large when the Hessian matrix is nearly singular indicating that the model may not well identified.)

The fixed effects estimates are given in the familiar format used in all of Stata's estimation commands. The variance of the level 2 (id) random effect is estimated as 1.675 with a standard error of 0.662.

We normally recommend that `gllamm` should be used with the `trace` option. This provides more information on how the syntax has been interpreted and shows the iterations of the maximum likelihood procedure. Using this option makes it easier to assess how long the program will take and to make sure that the model has been specified correctly. Using the above command with the `trace` option, gives the following output:

```
. xi: gllamm dtlm i.group level, i(id) family(binom) link(logit) nip(20) trace
i.group           _Igroup_1-3         (naturally coded; _Igroup_1 omitted)

General model information
------------------------------------------------------------------------------
dependent variable:         dtlm
family:                     binom
link:                       logit
denominator:                1
equation for fixed effects  _Igroup_2 _Igroup_3 level _cons

Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (id) equation(s):

   standard deviation of random effect
   id: _cons

number of level 1 units = 677
number of level 2 units = 226

Initial values for fixed effects


Iteration 0:   log likelihood = -373.67941
Iteration 1:   log likelihood = -317.84501
Iteration 2:   log likelihood = -313.96138
Iteration 3:   log likelihood = -313.89083
Iteration 4:   log likelihood = -313.89079

Logit estimates                                   Number of obs   =        677
                                                  LR chi2(3)      =     119.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -313.89079                       Pseudo R2       =     0.1600

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Igroup_2 |  -.1396641   .2282452    -0.61   0.541    -.5870164    .3076882
   _Igroup_3 |  -.8313329   .2742339    -3.03   0.002    -1.368821   -.2938444
       level |  -1.313382   .1409487    -9.32   0.000    -1.589636   -1.037127
       _cons |  -1.160498   .1824502    -6.36   0.000    -1.518094   -.8029024
------------------------------------------------------------------------------

start running on 16 Apr 2001 at 09:56:59

------------------------------------------------------------------------------
Iteration 0:
Coefficient vector:
         dtlm:      dtlm:      dtlm:      dtlm:        id:
```

```
    _Igroup_2  _Igroup_3      level      _cons      _cons
r1  -.1396641  -.8313329  -1.313382  -1.160498         .5

                                                log likelihood = -310.89641
------------------------------------------------------------------------------
Iteration 1:
Coefficient vector:
         dtlm:      dtlm:      dtlm:      dtlm:        id:
    _Igroup_2  _Igroup_3      level      _cons      _cons
r1  -.1511037  -.9209352  -1.478316   -1.31004   1.246432

                                                log likelihood = -306.86363
------------------------------------------------------------------------------

>>>> More iterations and final output
```

The information that the equation for the fixed effects is `_Igroup_2 _Igroup_3 level _cons` means that these are the column names of the parameter vector printed out in the iteration log. The initial output also tells us that the standard deviation for the random effect has equation name `id` and column name `_cons` in the parameter vector. (The absolute value of this parameter should be interpreted as the standard deviation; it can also be negative.)

The initial values for the fixed effects are estimated using conventional logistic regression. An arbitrary value of 0.5 is then used as the initial estimate of the standard deviation of the random intercept. In the first iteration, this changes to 1.246. We can watch the change in parameter values and log-likelihood until convergence.

We now consider a model that cannot be estimated in `xtlogit` by introducing a random effect for families. This gives a three level logistic regression model as in equation (3.1). Here the level 3 identifier, `famnum`, is simply specified within the `i()` option:

```
. xi: gllamm dtlm i.group level, i(id famnum) family(binom) link(logit) nip(8)
i.group           _Igroup_1-3         (naturally coded; _Igroup_1 omitted)

number of level 1 units = 677
number of level 2 units = 226
number of level 3 units = 118

Condition Number = 4.2148989

gllamm model

log likelihood = -305.11873

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Igroup_2 |      -.249   .3544871    -0.70   0.482     -.943782     .445782
   _Igroup_3 |  -1.052334   .3999883    -2.63   0.009    -1.836297   -.2683716
       level |  -1.648503   .1932191    -8.53   0.000    -2.027206   -1.269801
       _cons |  -1.485952   .2848916    -5.22   0.000    -2.044329   -.9275745
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)
```

```
    var(1): 1.1345171 (.68948027)

***level 3 (famnum)

    var(1): .57347062 (.53124401)
------------------------------------------------------------------------------
```

The variance at level 2 is estimated as 1.134 with a standard error of 0.689, and the variance at level 3 is estimated as 1.134 with a standard error of 0.689. The output only shows the final results of the estimation. Again, we would normally use the `trace` option to check that the model was correctly specified and to obtain the full iteration log. To check if 8 points were sufficient, we can use the commands

```
matrix a=e(b)
xi: gllamm dtlm i.group level, i(id famnum) family(binom) link(logit) nip(20)
  */ from(a) trace
```

Here, the vector of parameter estimates is stored in the matrix `a` and then passed to `gllamm` as intial values using the `from()` option. Using 20 quadrature points (which takes a long time since evaluation of the marginal likelihood requires summing 20×20 terms), some of the parameter estimates change in the third decimal place. Increasing the number of quadrature points by another 10 gives negligible changes.

Using the `eform` option when estimating the model or issuing the command `gllamm, eform` after estimating the model gives the same output as above but with exponentiated estimates and confidence intervals in the fixed-effects table.

See Section 8.3 for an example of three level ordinal logistic regression.

3.2 Two level random coefficient model

Here we illustrate the use of random coefficient models for normally distributed responses. Note, however, that we would normally not recommend 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.

The data for this section are the Junior School Project data from the MLn manual (Woodhouse, 1995). Maths results are available on pupils from different schools in the third and fifth years. We will fit a linear regression model of the year 5 results, `math5`, on the (mean centred) year 3 results, `math3`, with a random intercept and a random coefficient of `math3` for schools. The model can be written as

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + u_{0i} + u_{1i} x_{ij} \tag{3.2}$$

where i indexes the schools and j indexes the pupils, x_{ij} is the year 3 result and u_{1i} is the corresponding random coefficient. The two random effects are assumed to have a bivariate normal distribution.

3.2.1 Data preparation

A listing of the variables in the file *jsp.dta* is shown below for observations 87 to 95.

```
. list school pupil math5 math3 wt1 in 87/95
       school      pupil      math5      math3          wt1
 87.        5         21         28        3.6            1
 88.        5         22         30       -3.4            1
 89.        5         23         25       -3.4            1
 90.        5         24         37        6.6            2
 91.        5         25         36        1.6            1
 92.        6          1         28       -5.4            1
 93.        6          2         26        4.6            1
 94.        6          3         30       -6.4            1
 95.        6          4         37        5.6            1
```

The variable `wt1` contains level 1 weights and is equal to 1 for most pupils because there were only a few instances of two pupils in the same school having the same result for `math3` and `math5`.

3.2.2 Model fitting

When any of the random effects are not intercepts, we must specify the sets of variables whose linear combination multiplies the random effects. This is done by defining equations prior to running `gllamm` using the `eq` command (still available in Stata 7 but undocumented). The syntax is

$$\text{eq } \textit{name}\text{: } \textit{var1 var2 var3} \cdots$$

The equation can now be referred to by its name and the variables on the right hand side will be combined to form a linear combination $\boldsymbol{\lambda}'\mathbf{z}$ as in equation (1.2). In the `gllamm` command, we must first specify that there are two random effects at level 2 using the `nrf()` option. We then use the `eqs()` option to specify that one of the random effects, the random intercept, multiplies a variable, `cons`, equal to 1 and the other random effect, the random coefficient, multiplies `math3`.

Initially, we will assume zero correlation between the random slope and intercept by using the `nocor` option. The `weight()` option is used to inform `gllamm` that the data are in collapsed form and that `wt1` represents frequency weights for the level 1 units. (`gllamm` will also look for a `wt2` variable, but if this is not found, as here, the level 2 weights will be assumed to be equal to 1.)

```
. gen cons = 1
. eq sch_c: cons
. eq sch_m3: math3
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nocor nip(8) weight(wt)

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 12.01191

gllamm model

log likelihood = -2763.3492

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
```

```
------------+----------------------------------------------------------------
       math3 |   .6137155   .0443395    13.84   0.000     .5268117    .7006192
       _cons |   30.68167   .2952553   103.92   0.000     30.10298    31.26036
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  26.935556 (1.357001)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (school)

    var(1): 4.0987663 (.99349648)
    cov(1,2): fixed at 0

    var(2): .03747035 (.01981001)
------------------------------------------------------------------------------
```

The within-school (residual) level 1 variance is estimated as 26.94 with a standard error of 1.36

In order to interpret the level 2 variances, we need to consider that, in the `eqs()` option, equation `sch_c` was specified first, followed by equation `sch_m3`. Therefore, the first random effect is the random intercept (it multiplies `cons`) and the second random effect to be the random slope of `math3`. The random intercept variance, `var(1)`, is therefore estimated as 4.10 with a standard error of 0.99 and the random slope variance, `var(2)`, is estimated as 0.037 with a standard error of 0.020. The output reminds us that the covariance was fixed at 0.

Comparing the estimated variance of the random coefficient with its standard error (using a Wald test) gives the impression that it is not significant at the 5% level. However, the variance estimate is unlikely to be normally distributed and the Wald test is known to be invalid when the null value is on or near the boundary of the parameter space. A likelihood ratio test should therefore be used. The program estimates the Cholesky decomposition of the covariance matrix of the random effects. Since the covariance was set to 0, the Cholesky decomposition is diagonal with diagonal elements equal to the standard deviations (apart from the sign which may be negative). We can obtain the standard deviation estimates and their standard errors using the `allc` option:

```
. gllamm, allc

>>>> output as above, omitted

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
math5        |
       math3 |   .6137155   .0443395    13.84   0.000     .5268117    .7006192
       _cons |   30.68167   .2952553   103.92   0.000     30.10298    31.26036
-------------+----------------------------------------------------------------
lns1         |
       _cons |   1.646724   .0251898    65.37   0.000     1.597353    1.696095
-------------+----------------------------------------------------------------
scho1        |
        cons |   2.024541   .2453634     8.25   0.000     1.543638    2.505444
-------------+----------------------------------------------------------------
scho2        |
       math3 |  -.1935726   .0511695    -3.78   0.000    -.2938629   -.0932823
------------------------------------------------------------------------------
```

The standard deviation of the random coefficient is estimated as 0.193 with a standard error of 0.05 - this would be significant at the 5% level. (`[lns1]_cons` is the log standard deviation at level 1.)

In order to carry out the likelihood ratio test, we will now fit the model without a random coefficient for `math3`. It will be quicker to use as starting values the previous estimates, obtained using `e(b)`, and passed to `gllamm` using the `from()` and `skip` options. Since the parameter vector has equation name `scho1` and column name `cons` for the intercept standard deviation estimate (see above), we will use the option `eqs(sch_c)` although this would normally not be necessary for a random intercept model:

```
. matrix a=e(b)
. gllamm math5 math3, i(school) nip(20) eqs(sch_c) weight(wt) from(a) skip

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 14.000586

gllamm model

log likelihood = -2767.866

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6087711   .0326381    18.65   0.000     .5448016    .6727406
       _cons |   30.60566   .3407557    89.82   0.000     29.93779    31.27353
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  28.118708 (1.3728251)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (school)

    var(1): 4.0556597 (1.2020057)
------------------------------------------------------------------------------
```

The likelihood ratio test is

```
. disp chiprob(1,2*(2767.866 -2763.3492))
.00265062
```

showing that the effect of `math3` varies significantly between schools.

We now introduce a correlation between the random slope and intercept, using as starting values the estimates from the model with uncorrelated slope and intercept (these estimates are still in the vector `a`):

```
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(8) weight(wt) from(a)

number of level 1 units = 887
```

```
number of level 2 units = 48

Condition Number = 13.579083

gllamm model

log likelihood = -2757.0046

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6073693    .042477    14.30   0.000      .524116    .6906226
       _cons |   30.65958   .3311732    92.58   0.000     30.01049    31.30867
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  26.934382 (1.3527423)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (school)

    var(1): 4.1463278 (.95033318)
    cov(1,2): -.31583208 (.09461644) cor(1,2): -.86902604

    var(2): .03185542 (.01655613)
------------------------------------------------------------------------------
```

The intercept and slope are highly negatively correlated (correlation = -0.869). The difference in log-likelihoods indicates that this correlation is highly significant.

Again, the Cholesky decomposition of the covariance matrix, $\mathbf{L}$, was estimated and the covariance matrix and corresponding standard errors were computed from $\mathbf{L}$ and its standard errors. To view these 'raw' parameters, use the `allc` option. Since the parameters have just been estimated, we can simply redisplay them in their raw form using the command `gllamm, allc`.

We will now use adaptive quadrature to improve the parameter estimates. We would recommend always trying adaptive quadrature if the responses are continuous (or perhaps for any types of responses). Simply run the same command as before but with the `adapt` option.

```
. matrix a=e(b)
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(8) weight(wt) from(a) adapt

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 14.9902

gllamm model

log likelihood = -2757.0803

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6123977   .0428954    14.28   0.000     .5283242    .6964712
       _cons |   30.59295   .3655917    83.68   0.000      29.8764    31.30949
```

```
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  26.964585 (1.3460859)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (school)

    var(1): 4.5788594 (1.3101581)
    cov(1,2): -.3605893 (.12256045) cor(1,2): -.90984435

    var(2): .03430316 (.01760728)
------------------------------------------------------------------------------
```

These estimates are close to those obtained using MLwiN, namely fixed effects (standard error): 0.6124 (0.04283) and 30.59 (0.3657), level 1 residual variance: 26.96 (1.343) and covariance matrix of the random effects:

$$\begin{bmatrix} 4.585\ (1.291) & -0.3603\ (0.1189) \\ -0.3606\ (0.1189) & 0.03423\ (0.01704) \end{bmatrix}.$$

We can use `gllapred` to obtain the posterior means (empirical Bayes predictions) of the random effects:

```
. gllapred u,u
Non-adaptive log-likelihood: -2757.1446
-2757.5280 -2757.0861 -2757.0803 -2757.0803
log-likelihood is -2757.0803
```

This creates four variables, `um1` and `um2` containing the posterior means of the random intercept and coefficient, respectively, and `us1` and `us2` containing the corresponding posterior standard deviations. (The final log-likelihood value returned by `gllapred` should be the same as that returned by `gllamm`, otherwise there is a problem!)

Values of the posterior means are created for each observation in each school. This can be seen by listing the same observations as before,

```
. list school pupil  um1 um2 us1 us2 in 87/95

       school      pupil         um1         um2         us1         us2
 87.        5         21  -.03652747  -.00149414    1.044302   .10956977
 88.        5         22  -.03652747  -.00149414    1.044302   .10956977
 89.        5         23  -.03652747  -.00149414    1.044302   .10956977
 90.        5         24  -.03652747  -.00149414    1.044302   .10956977
 91.        5         25  -.03652747  -.00149414    1.044302   .10956977
 92.        6          1   .85082254  -.07551827   1.1132732   .11470399
 93.        6          2   .85082254  -.07551827   1.1132732   .11470399
 94.        6          3   .85082254  -.07551827   1.1132732   .11470399
 95.        6          4   .85082254  -.07551827   1.1132732   .11470399
```

We can summarise the values of `um1` and `um2` for the 48 schools by first creating a dummy variable, `f`, that is equal to one for only one observation in each school:

```
. sort school pupil
. qui by school: gen f=_n==1
. summ um1 um2 if f==1

    Variable |     Obs        Mean   Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         um1 |      48   -1.58e-08   1.853219  -3.834251   3.270771
         um2 |      48    2.65e-09   .1517105  -.2679363   .3366223


. corr um1 um2 if f==1, cov
(obs=48)

             |      um1      um2
-------------+------------------
         um1 |  3.43442
         um2 | -.277808   .023016
```

The variances of the posterior means are smaller than the estimated variances of the (prior) distribution of the random effects. This is because the posterior means are 'shrunken' towards the mean of the prior distribution, in this case 0. They are therefore sometimes referred to as *shrinkage estimators*.

We can use `gllapred` with the `linpred` option to get predicted values for each individual child based on

$$\begin{aligned} \hat{y}_{ij} &= \hat{\eta}_{ij} \\ &= \hat{\beta}_0 + \hat{\beta}_1 x_{ij} + \hat{u}_{0i} + \hat{u}_{1i} x_{ij}, \end{aligned} \tag{3.3}$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the fixed parameter estimates and $\hat{u}_{0i}$ and $\hat{u}_{1i}$ are the posterior means of the random intercept and slope, respectively.

```
. gllapred lp, linpred
Non-adaptive log-likelihood: -2757.1446
-2757.5280 -2757.0861 -2757.0803 -2757.0803
log-likelihood is -2757.0803
```

We can plot the predictions agains `math3` for each school by first sorting the data by `math3` within `school` and then using the `connect(L)` option which connects only groups of points for which `math3` increases:

```
. sort school math3
. graph lp math3, connect(L) s(i) xlab ylab l1(predicted math5 score) b2(math3 score) gap(3)
```

The resulting graph is shown in Figure 3.1.

Level 1 residuals could be computed by subtracting `lp` from `math5`. The empirical Bayes estimates are sometimes considered level 2 residuals. Outlying schools could be detected by dividing the level 2 residuals by the standard errors (the posterior standard deviations).

See also Section 5.2 for a discrete random effects version of this model.

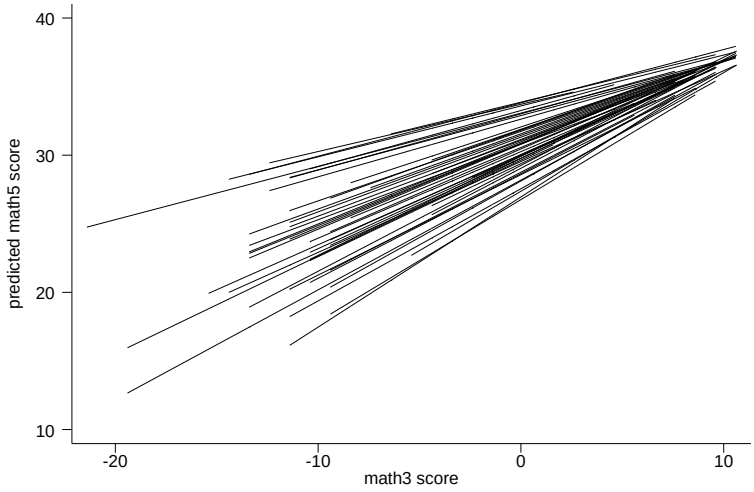


Figure 3.1: Predictions from random coefficients model for JSP data.

Chapter 4

Multilevel factor models

4.1 One parameter and two parameter item-response models

Binary data on 5 items from section 6 of the Law School Admission Test (LSAT)(Bock and Lieberman, 1970), will be used to illustrate how factor models may be fitted using `gllamm`.

Item response models may be used to model the responses of subjects to a number of exam questions, or items. The log odds of subject i giving a correct answer to item j may be modelled using a one parameter logistic item response model (Rasch model):

$$\eta_{ij} = \beta_j + u_i \tag{4.1}$$

where $-\beta_j$ represents the difficulty of question j and u_i represents the ability of subject i. This is a simple two-level model and may be fitted using `xtlogit`. If we introduce a further parameter λ_j, we obtain a two parameter logistic item response model

$$\eta_{ij} = \beta_j + u_i\lambda_j \tag{4.2}$$

where λ_j represents the extent to which question j discriminates between subjects of different abilities. Here we are modelling a multivariate dataset by representing the variables as level 1 units indexed j. If the data are in long form with the responses to all the questions stacked into a single response vector, we can use dummy variables

$$x_{pij} = \begin{cases} 1 \text{ if p=j} \\ 0 \text{ otherwise} \end{cases} \tag{4.3}$$

to write the model in the form of equation (1.2), giving

$$\eta_{ij} = \boldsymbol{\beta}'\mathbf{x}_{ij} + u_i\boldsymbol{\lambda}'\mathbf{x}_{ij} \tag{4.4}$$

(See Section 1.1.2 for more details on defining factor models.)

4.1.1 Data preparation

The data are in *lsat.dta*. The responses are stacked into the variable `resp` and the variables `i1` to `i5` are indicators for the 5 items. Here we list some of these variables for observations 1 to 10.

```
. list id resp wt2 i1 i2 i3 in 1/10

        id       resp        wt2        i1        i2        i3
```

```
  1.        1         0         3         1         0         0
  2.        1         0         3         0         1         0
  3.        1         0         3         0         0         1
  4.        1         0         3         0         0         0
  5.        1         0         3         0         0         0
  6.        2         0         6         1         0         0
  7.        2         0         6         0         1         0
  8.        2         0         6         0         0         1
  9.        2         0         6         0         0         0
 10.        2         1         6         0         0         0
```

The values in the variable `wt2` are level 2 weights and give the number of subjects with the same response pattern across the 5 items.

4.1.2 Model fitting

A simple one parameter logistic (or Rasch) model (see equation (4.1)) may be fitted using

```
. gllamm resp i1 i2 i3 i4 i5, nocons link(logit) fam(bin) i(id) nip(10) /*
>  */ weight(wt)

number of level 1 units = 5000
number of level 2 units = 1000

Condition Number = 2.3633019

gllamm model

log likelihood = -2466.9376

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          i1 |   2.730013    .130441    20.93   0.000     2.474353    2.985673
          i2 |   .9986051   .0791772    12.61   0.000     .8434207     1.15379
          i3 |   .2398536   .0717746     3.34   0.001     .0991779    .3805292
          i4 |    1.30645    .084638    15.44   0.000     1.140563    1.472338
          i5 |   2.099404   .1054449    19.91   0.000     1.892736    2.306072
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .57022544 (.10486119)
------------------------------------------------------------------------------
```

The random effects variance is estimated as 0.570 with a standard error of 0.105. (Note that the same model may be fitted using `xtlogit resp i1-i5, nocons i(id) quad(10)` if the data is not in 'collapsed' form.)

In order to fit a two-parameter item-response model in equation (4.4), we first need to define an equation using the `eq` command and then use the `eqs()` option to specify the variables `i1` to `i5` in the linear combination of variables that multiplies the latent variable in (4.4).

```
. eq id: i1 i2 i3 i4 i5
. gllamm resp i1 i2 i3 i4 i5, nocons link(logit) family(binom) i(id)/*
>  */ eqs(id) nip(10) w(wt) lf0(6 -2466.9376252) trace

number of level 1 units = 5000
number of level 2 units = 1000

Condition Number = 11.496849



gllamm model                                        Number of obs   =       1000
                                                    LR chi2(4)      =       0.57
Log likelihood = -2466.6534                         Prob > chi2     =     0.9665

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          i1 |   2.773176   .2056812    13.48   0.000     2.370049    3.176304
          i2 |   .9902013   .0900178    11.00   0.000     .8137696    1.166633
          i3 |   .2491494   .0762736     3.27   0.001     .0996559    .3986429
          i4 |    1.28476   .0990366    12.97   0.000     1.090651    1.478868
          i5 |   2.053274   .1353578    15.17   0.000     1.787978     2.31857
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .68157927 (.42601805)

    loadings for random effect 1
    i2: .87541638 (.36267338)
    i3: 1.0790915 (.43509518)
    i4: .83378521 (.36723728)
    i5: .79561451 (.38058924)

------------------------------------------------------------------------------
```

Using the `lf0()` option caused the likelihood ratio test (`LR chi2(4) = 0.57`) to be shown which indicates that the factor loadings do not differ significantly from 1, i.e. the one parameter item response model was adequate.

Here the factor loading of item 1 was constrained to 1 and the variance of the random effect was estimated freely. To obtain the parameters of the model where the standard deviation is constrained to 1 instead, we can interpret the standard deviation `sqrt(.87541533)` as the first loading and multiply all other loadings by this value. See Section 8.2 for a discussion of item-response models for ordinal data.

Chapter 5

Discrete random effects

5.1 A simple finite mixture model

In the Handbook of Statistical Analyses using Stata, Rabe-Hesketh and Everitt (2000) describe finite mixture modelling using Stata's `ml` functions. Here we will use `gllamm` to fit a simple finite mixture model the age of onset of schizophrenia data used in the book.

According to the subtype model of schizophrenia, there are two types of schizophrenia. One is characterized by early onset, typical symptoms and poor premorbid competence and the other by late onset, atypical symptoms and good premorbid competence. We will investigate this question by fitting a mixture of two normal distributions to the ages. (If we had variables on symptoms and premorbid competence, we could fit a more general latent class model.) The finite mixture model can be written as

$$f(y_i;\pi_1,\mu_1,\mu_2,\sigma_1,\sigma_2) = \pi_1 g(y_i;\mu_1,\sigma_1) + (1-\pi_1)g(y_i;\mu_2,\sigma_2) \tag{5.1}$$

where $g(y;\mu,\sigma)$ is the Gaussian density with mean μ and standard deviation σ,

$$g(y;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right\} \tag{5.2}$$

and π_1 and π_2 are the mixing probabilities.

To write this model as a GLLAMM, we constrain $\sigma_1 = \sigma_2 = \sigma$. Conditional on u_i, y_i has a normal distribution with variance σ^2 and expectation

$$E[y_i|u_i] = \eta_i \tag{5.3}$$

where

$$\eta_i = \mu + u_i \tag{5.4}$$

and u_i is a discrete latent variable with two values, $z_1 = \mu_1 - \mu$ and $z_2 = \mu_2 - \mu$ where μ is the overall mean.

5.1.1 Data preparation

First we read the data

```
infile ages using onset.dat, clear
```

We need to specify the 'level 2 units' in `gllamm`, i.e. the units i over which u_i varies, in this case the subjects:

```
gen id=_n
```

The first ten ages are:

```
. list id ages in 1/10

            id       ages
  1.         1         20
  2.         2         30
  3.         3         21
  4.         4         23
  5.         5         30
  6.         6         25
  7.         7         13
  8.         8         19
  9.         9         16
 10.        10         25
```

5.1.2 Parameter estimation

Since sensible starting values are crucial for finite mixture models, we will first estimate the one class solution and then use the Gateaux derivative method to introduce a second class. Having both solutions will also enable us to assess the change in log-likelihood although the log-likelihood ratio test is strictly not valid for mixtures.

We can estimate the one class solution using the `ip(f)` option and by specifying one integration point using the `nip()` option:

```
gllamm ages, i(id) ip(f) nip(1)
```

In order to force `gllamm` to estimate the parameters by maximum likelihood rather than ordinary regression (for comparison with the two class solution), we must make the problem look like a nonstandard regression problem. One possibility is to use the `s()` option which allows the residual variance to vary with covariates - here we will use a 'covariate' equal to 1.

```
. gen cons = 1
. eq het: cons
. gllamm ages, i(id) ip(f) nip(1) s(het) trace

General model information
------------------------------------------------------------------------------
dependent variable:         ages
family:                     gauss
link:                       ident
equation for fixed effects  _cons

Random effects information for 2 level model
------------------------------------------------------------------------------
***level 1 equation:

   log standard deviation
   lns1: cons

>>> output omitted
```

```
                                                  log likelihood = -383.39585
------------------------------------------------------------------------------
------------------------------------------------------------------------------
        ages |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ages         |
       _cons |   30.47475   1.169045    26.07   0.000     28.18346    32.76603
-------------+----------------------------------------------------------------
lns1         |
        cons |   2.453747   .0710669    34.53   0.000     2.314458    2.593035
------------------------------------------------------------------------------
```

The log standard deviation is estimated as 2.453747 and the log-likelihood is -383.39585.

We can now use the `gateaux()` option, `gateaux(min max num)`, to introduce another mass. Setting all the parameters equal to the estimates of the one class solution (obtained using `e(b)`), a very small new mass will be moved from `min` to `max` in `num` steps. If the log-likelihood increases at any of these locations, a new mass is introduced at the location with the largest increase in log-likelihood and all parameters are updated to maximise the log-likelihood for the two class solution. We need to pass the current log-likelihood to `gllamm` using the `lf0()` option. The syntax is `lf0(k ll)` where `k` is the number of parameters of the current solution and `ll` is the log-likelihood of the current solution:

```
. matrix a=e(b)
. local ll=e(ll)
. local k=e(k)
. gllamm ages, i(id) ip(f) nip(2) s(het) lf0(`k' `ll') /*
>  */ gateaux(-20 20 100) from(a) trace

>>> output omitted

number of level 1 units = 99
number of level 2 units = 99

Condition Number = 23.174028



gllamm model                                      Number of obs   =         99
                                                  LR chi2(2)      =      19.40
Log likelihood = -373.69749                       Prob > chi2     =     0.0001

------------------------------------------------------------------------------
        ages |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   30.47475   1.169045    26.07   0.000     28.18346    32.76603
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  44.646412 (7.8530966)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (id)
    prob: 0.2515, 0.7485
```

```
    loc1: 16.426, -5.5187
  var(1): 90.653445
------------------------------------------------------------------------------
```

We therefore have a mixture component (or latent class) with a small mixing probability (or prior probability) estimated as 0.25 whose mean age of onset is 16.43 years greater than the overall average age of onset of 30.47 and a larger class (estimated prior probability of 0.75) whose age of onset is -5.52 years lower than the overall average. In `gllamm` the variance must be assumed to be equal in both classes and is estimated as 44.65. (Allowing different variances in the classes as in Rabe-Hesketh and Everitt (2000), hardly increases the log-likelihood, so for these data the assumption of equal variances is appropriate.)

To see how the model is parameterised, use the `allc` option:

```
. gllamm, allc

>>> same output as without allc option (omitted)


gllamm model                                    Number of obs   =         99
                                                LR chi2(2)      =      19.40
Log likelihood = -373.69749                     Prob > chi2     =     0.0001

------------------------------------------------------------------------------
        ages |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
ages         |
       _cons |   30.47475   1.169045    26.07   0.000     28.18346    32.76603
-------------+----------------------------------------------------------------
lns1         |
        cons |   1.899387   .0879477    21.60   0.000     1.727013    2.071761
-------------+----------------------------------------------------------------
z2_1         |
       _cons |   16.42646   1.852972     8.86   0.000      12.7947    20.05822
-------------+----------------------------------------------------------------
p2_1         |
       _cons |  -1.090743   .2742904    -3.98   0.000    -1.628342   -.5531438
------------------------------------------------------------------------------
```

Here, `p2_1` is the log odds for class 1 so that the estimated probability is

$$\hat{\pi}_1 = \frac{\exp(\texttt{p2\_1})}{1+\exp(\texttt{p2\_1})} \tag{5.5}$$

and `z2_1` is the location for class 1

$$\hat{z}_1 = \mu_1 - \mu = \texttt{z2\_1}. \tag{5.6}$$

The location for class 2 is estimated as

$$\hat{z}_2 = \mu_2 - \mu = \hat{z}_1\hat{\pi}_1/(1-\hat{\pi}_1) \tag{5.7}$$

so that the mean of the discrete probability distribution of the latent variable is zero

$$\hat{z}_1\hat{\pi}_1 + \hat{z}_2(1-\hat{\pi}_1) = 0 \tag{5.8}$$

We can obtain estimates of the posterior probabilities and posterior means using the `gllapred` command:

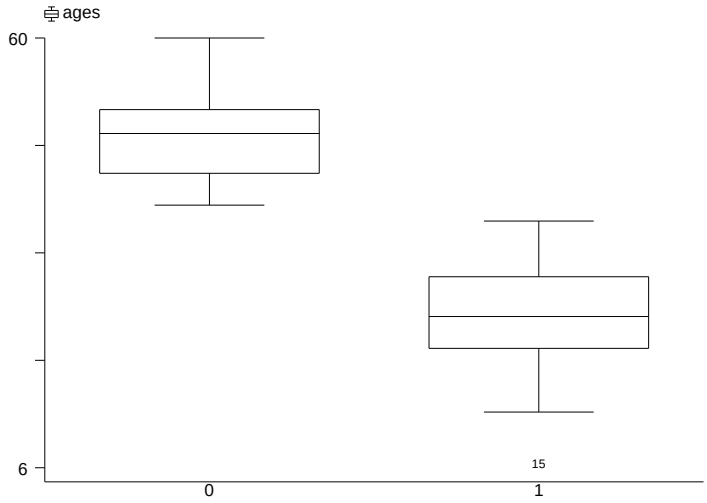


Figure 5.1: Boxplots of ages of onset by assigned latent class

```
. gllapred u, p
prior probabilities

M_zps2[1,2]
           c1         c2
r1  .25147836  .74852164

locations for random effect 1

M_zlc2[1,2]
            c1          c2
r1   16.426462  -5.5187445

log-likelihood is -373.69749
```

The output reminds us what the prior probabilities were and gives us the log-likelihood which should be identical to that given in the `gllamm` output. The posterior probabilities are stored in the variables `u1` and `u2`.

We can assign individuals to the class with the greatest posterior probability and produce boxplots of the ages within each class:

```
gen class = u2>u1
sort class
graph ages, by(class) box s([id])
```

giving the graph in Figure 5.1.

We could also use the `ip(fn)` option to estimate the means of the two latent classes directly rather than their deviation from a common mean. This is an example of a model with no fixed effects. For the one class solution, use:

```
. gllamm ages, nocons i(id) ip(fn) nip(1)  trace

>>> output omitted
```

```
number of level 1 units = 99
number of level 2 units = 99

Condition Number = 16.449916

gllamm model

log likelihood = -383.39585

No fixed effects

Variance at level 1
------------------------------------------------------------------------------
  135.29987 (19.230686)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (id)
    prob: 1

    loc1: 30.475
  var(1): 0
------------------------------------------------------------------------------
```

Then introduce another point using the `gateaux` option:

```
. matrix a=e(b)
. local ll=e(ll)
. local k=e(k)
. gllamm ages, nocons i(id) ip(fn) nip(2) lf0(`k' `ll') /*
>  */ gateaux(20 60 100) from(a) trace

>>> output omitted

number of level 1 units = 99
number of level 2 units = 99

Condition Number = 20.454287

gllamm model

log likelihood = -373.69749

No fixed effects

Variance at level 1
------------------------------------------------------------------------------
  44.646432 (7.8531028)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (id)
    prob: 0.2515, 0.7485

    loc1: 46.901, 24.956
  var(1): 90.653439
------------------------------------------------------------------------------
```

5.2 Linear mixed model with discrete random effects

In this section we will return to the Junior School Project data analysed in Section 3.2. Maths results are available on pupils from different schools in the third and fifth years. We will fit a linear regression model regressiong the year 5 results, `math5`, on the (mean centred) year 3 results, `math3`, with a random intercept and a random coefficient of `math3` for schools. The model can be written as

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + u_{0i} + u_{1i} x_{ij} \tag{5.9}$$

where i indexes the schools and j indexes the pupils, x_{ij} is the year 3 result and u_{1i} is the corresponding random coefficient. Instead of assuming a bivariate normal ditribution of the random effects as in Section 3.2, we now assume a bivariate discrete distribution, i.e., we assume that the random effects $(u_0,\ u_1)$ take on a number of discrete values $(z_{0r},\ z_{1r})$, with probabilities π_r, $r = 1, \cdots, R$. This corresponds to assuming that the population falls into a finite number of latent classes or types or can be approximated in this way. When the maximum number of classes is used, the distribution may be interpreted as a non-parametric distribution.

5.2.1 Model fitting

The data are in *jsp.dta*. The `gllamm` command is identical to that used in the continuous case except that the `ip(f)` option is specified. Initially we fit a model with just two points:

```
. use jsp, clear
. gen cons = 1
. eq sch_c: cons
. eq sch_m3: math3
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(2) weight(wt) ip(f)

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 25.665476

gllamm model

log likelihood = -2760.7033

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .5921935   .0397521    14.90   0.000     .5142807    .6701062
       _cons |   30.71605   .3245148    94.65   0.000     30.08001    31.35209
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  28.171487 (1.3495008)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (school)
    prob: 0.6727, 0.3273
```

```
    loc1: -1.2586, 2.5874
  var(1): 3.2566284
cov(1,2): -.285505

    loc2: .11034, -.22684
  var(2): .02502991
------------------------------------------------------------------------------
```

The coordinates of the two points are (intercept, slope) = (-1.2586,0.11034) and (2.5874,-0.22684) with probabilities of 0.6727 and 0.3273 respectively. The output also gives the variances and covariance of the discrete random effects based on the bivariate discrete probability distribution.

We can use the Gateaux derivative method to check if introduction of a further mass-point yields a larger maximised likelihood. Keeping all other parameters at their current values, we need to move a small mass through a fine 2-D grid of values of the random effects and check whether this increases the likelihood anywhere. We can do this using the `gateaux()` option to specify the limits and number of steps for the search in each dimension. In addition, we have to pass the number of parameters and log-likelihood of the current model to `gllamm` using the `lf0()` option. After finding the maximum Gateaux derivative point, the estimation of the extended model automatically starts of the Gateaux derivative is positive.

```
. matrix a=e(b)
. local ll=e(ll)
. local k=e(k)
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(4) weight(wt)
>  */ ip(f) from(a) gateaux(-10 10 30) lf0(`k' `ll')

maximum gateaux derivative is 1.491291

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 60.948279



gllamm model                                    Number of obs   =         48
                                                LR chi2(3)      =       6.52
Log likelihood = -2757.4437                     Prob > chi2     =     0.0889

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6015936   .0452026    13.31   0.000     .5129981    .6901891
       _cons |   30.70774   .3272301    93.84   0.000     30.06638     31.3491
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  27.690161 (1.3291571)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (school)
    prob: 0.6551, 0.0291, 0.3158
```

```
    loc1: -1.1642, -2.6896, 2.6625
  var(1): 3.3371899
cov(1,2): -.33125197

    loc2: .07502, .91947, -.24032
  var(2): .04652579
------------------------------------------------------------------------------
```

The likelihood ratio test produced at the top of the output is not strictly valid for comparing solutions with different numbers of masses but is printed whenever the `lf0()` option is used. A very small mass of 0.0291 has been placed at (-2.6896,.91947) without affecting the other masses substantially. Note that the condition number is quite large. This seems to happen frequently when a larger number of free masses are estimated. We now use the Gateaux derivative again to see if a fourth point can be introduced:

```
. matrix a=e(b)
. local ll=e(ll)
. local k=e(k)
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(4) weight(wt)
>  */ ip(f) from(a) gateaux(-10 10 30) lf0(`k' `ll')

maximum gateaux derivative is .27726546

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 55.415951



gllamm model                                        Number of obs   =         48
                                                    LR chi2(3)      =      12.57
Log likelihood = -2751.1611                         Prob > chi2     =     0.0057

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |    .616916   .0457258    13.49   0.000      .527295    .7065369
       _cons |   30.65184    .361109    84.88   0.000     29.94408     31.3596
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  26.644581 (1.2922193)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (school)
    prob: 0.5334, 0.0316, 0.1597, 0.2753

    loc1: -.34239, -2.6291, -3.3806, 2.9257
  var(1): 4.4623476
cov(1,2): -.34745907

    loc2: .06313, .88985, .08057, -.27122
  var(2): .04845221
------------------------------------------------------------------------------
```

The variances and covariance of the discrete random effects are now quite close to the estimates of the model assuming continuous random effects in Section 3.2. Note that a point with a substantial probability of 0.1597 has now been included. The previous three point solution may represent a local maximum of the log-likelihood since it seems likely that a better three point solution can be achieved by using as starting values the above estimates excluding the second point with the very low probability of 0.0316. We can do this as follows:

```
. matrix a=e(b)
. matrix list a

a[1,12]
         math5:      math5:       lns1:     z2_1_1:     z2_2_1:       p2_1:
         math3       _cons       _cons        cons       math3       _cons
y1     .616916   30.651841   1.6412929  -.34239476    .0631312   .66134194

        z2_1_2:     z2_2_2:       p2_2:     z2_1_3:     z2_2_3:       p2_3:
          cons       math3       _cons        cons       math3       _cons
y1  -2.6291197   .88984924  -2.1641927  -3.3806175   .08056978  -.54491932

. matrix b = a[1,1..6],a[1,10..12]
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(3) from(b) copy weight(wt) ip(f)

number of level 1 units = 887
number of level 2 units = 48

Condition Number = 37.366118

gllamm model

log likelihood = -2753.4705

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6071675   .0401361    15.13   0.000     .5285023    .6858327
       _cons |   30.62459   .3660526    83.66   0.000     29.90714    31.34204
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  27.002646 (1.3044171)

Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (school)
    prob: 0.539, 0.1851, 0.2759

    loc1: -.32603, -3.4409, 2.9461
  var(1): 4.6435926
cov(1,2): -.33181452

    loc2: .07632, .16678, -.26104
  var(2): .02708821
------------------------------------------------------------------------------
```

giving a higher maximised likelihood than previously. Here we used the `copy` option to make `gllamm` ignore the equation and column names of the matrix `b`.

We now return to the four class model by running

```
. gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(4) from(a) copy weight(wt) ip(f)
```

We can use `gllapred` to estimate the posterior means and probabilities of the random effects. First we estimate the posterior probabilities using the `p` option:

```
. gllapred z,p
prior probabilities

M_zps2[1,4]
           c1         c2         c3         c4
r1  .53340462   .0316186   .1596556  .27532118

locations for random effect 1

M_zlc2[1,4]
            c1          c2          c3          c4
r1  -.34239476  -2.6291197  -3.3806175   2.9256688

locations for random effect 2

M_zlc3[1,4]
            c1          c2          c3          c4
r1    .0631312   .88984924   .08056978  -.27122389


log-likelihood is -2751.1611
```

the output reminds us what the prior probabilities of class membership are and gives us the log-likelihood of the model which should be identical to that obtained previously using `gllamm`. The `p` option causes `gllapred` to compute the four posterior probabilities for each observation and store them in `z1` to `z4`.

First, we calculate the greatest probability

```
. egen double maxp = rmax(z1 z2 z3 z4)
. summ maxp

    Variable |     Obs        Mean   Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        maxp |     848    .8940566   .1446485   .5110811   .9998173
```

We can now classify the schools into four latent groups (or classes) by allocating them to the group with the largest posterior probability. For each school, there is a latent group to which the school belongs with a posterior probability of at least 51% (the minimum of `maxp`).

```
gen class = 1 if z1>=maxp
replace class = 2 if z2>=maxp
replace class = 3 if z3>=maxp
replace class = 4 if z4>=maxp
```

The posterior means are obtained in the same way as for continuous random effects:

```
. gllapred z,u

log-likelihood is -2751.1611
```

giving posterior means stored in zm1 and zm2 and posterior standard deviations in zs1 and zs2:

```
. list school pupil  zm1 zm2 zs1 zs2 in 87/95

       school      pupil         zm1         zm2         zs1         zs2
 87.        5         21  -.28267993     .055835     .521296   .04961322
 88.        5         22  -.28267993     .055835     .521296   .04961322
 89.        5         23  -.28267993     .055835     .521296   .04961322
 90.        5         24  -.28267993     .055835     .521296   .04961322
 91.        5         25  -.28267993     .055835     .521296   .04961322
 92.        6          1   .59825982  -.03396211   1.4983812   .15183799
 93.        6          2   .59825982  -.03396211   1.4983812   .15183799
 94.        6          3   .59825982  -.03396211   1.4983812   .15183799
 95.        6          4   .59825982  -.03396211   1.4983812   .15183799
```

We can estimate the same model assuming normally distributed random effects (Section 3.2) and obtain the corresponding posterior means:

```
gllamm math5 math3, i(school) nrf(2) eqs(sch_c sch_m3) nip(8) weight(wt) adapt
gllapred u, u
```

giving posterior means um1 and um2. We can tabulate the mean posterior means from the continuous model for each latent group of the discrete model:

```
. sort school pupil
. qui by school: gen f=_n==1
. table class if f==1, contents(mean um1 mean um2 freq)

----------------------------------------------
    class |  mean(um1)   mean(um2)       Freq.
----------+-----------------------------------
        1 | -.21519023   .02161626          28
        2 | -3.6618951   .33129555           1
        3 | -2.7079715   .20277729           7
        4 |  2.3740233  -.19526499          12
----------------------------------------------
```

The covariance matrices of the two sets of posterior means are

```
. corr zm1 zm2 if f==1, cov
(obs=48)

             |      zm1      zm2
-------------+------------------
         zm1 |  3.56467
         zm2 | -.280939  .037933


. corr um1 um2 if f==1, cov
(obs=48)
```

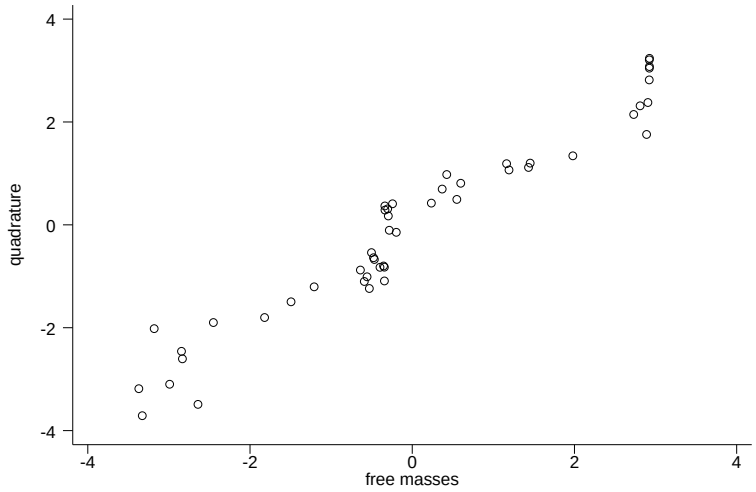


Figure 5.2: Posterior means of random intercept: quadrature solution versus discrete solution

```
             |      um1      um2
-------------+------------------
         um1 |  3.44322
         um2 | -.278019   .022844
```

These are remarkably similar as are the model estimates of the covariance matrices of the random effects. Figure 5.2 and 5.3 show scatterplots of the two estimates of the random intercepts and of the two estimates of the random slopes, respectively. These figures (in encapsulated postscript) were created using the commands

```
graph um1 zm1 if f==1, xlab ylab l1(quadrature) b2(free masses)  gap(3) saving(jspres1, replace)
translate jspres1.gph jspres1.eps, trans(gph2eps)

graph um2 zm2 if f==1, xlab ylab l1(quadrature) b2(free masses) gap(3) saving(jspres2, replace)
translate jspres2.gph jspres2.eps, trans(gph2eps)
```

See Section 9.4 for another example of using discrete latent variables.

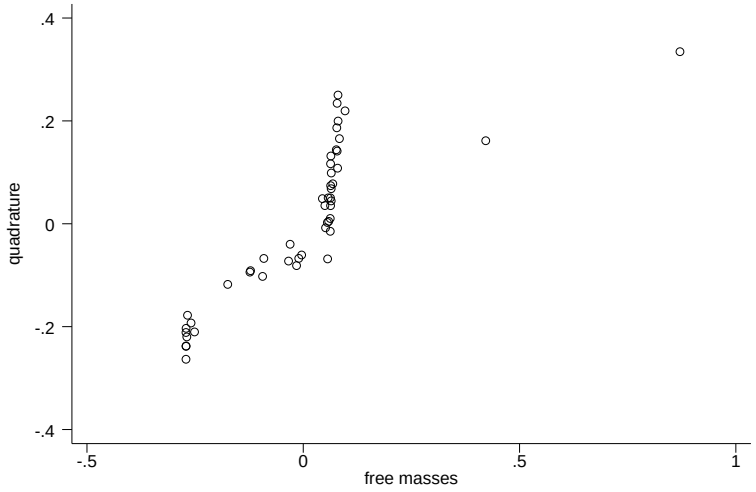


Figure 5.3: Posterior means of random slope: quadrature solution versus discrete solution

Chapter 6

Mixed response models

The models we have discusses so far have had responses of a single type, dichotomous, ordinal or continuous. A single link and family were specified. In this chapter we discuss models where the responses are of mixed types, for example dichotomous and continuous. For such models, different links and families are specified for different responses.

6.1 Logistic regression with covariate measurement error

An epidemiological dataset with variables on diet and coronary heart disease (CHD) (Morris *et al.*, 1977) will be used to illustrate how the program may be used for logistic regression with errors in covariates. The aim is to estimate the relationship between fibre intake (exposure) and risk of CHD (disease) where fibre is subject to measurement error and has been measured twice on a subset of subjects.

We therefore have several responses per subject, one or two of the fallible measure of exposure y_{i1}, y_{i2} and disease status y_{i3}. If we wish to model the relationship between exposure and disease status whilst correcting for measurement error in exposure, we need to specify a *measurement model* for exposure. We assume that that the exposure measurements y_{i1} and y_{i2} are independently normally distributed conditional on true exposure with means μ_{ij} given by

$$\begin{aligned} \eta_{ij} = \mu_{ij} &= \beta_j + u_i\lambda_j, \quad j = 1, 2 \\ &= \beta_1 + u_i \end{aligned} \tag{6.1}$$

where the mean exposure on both occasions is assumed to be the same ($\beta_2 = \beta_1$), u_i is a latent variable representing the difference between subject ith's exposure and the mean exposure, and we assume that the scale of both measurements is the same by setting $\lambda_1 = \lambda_2 = 1$. By consstraining the factor loadings to 1, we ensure that the scale of u_i is the same as that of the exposure measurements.

We now specify a *disease model* by assuming that y_{i3} is binomial with the logit of the probability π_{i3} given by

$$\eta_{i3} = \text{logit}(\pi_{i3}) = \beta_3 + u_i\lambda_3. \tag{6.2}$$

λ_3 is the log odds ratio of interest – the estimated log of the ratio of the odds of having the disease when the true exposure increases by one unit.

These models can be written as a GLLAMM model by using appropriate dummy variables,

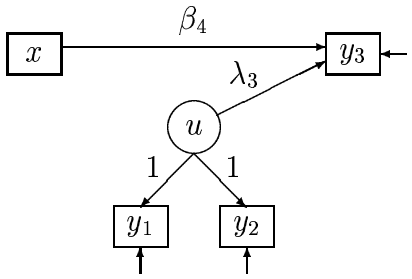
$$\begin{aligned} \eta_{ij} &= \beta_1 z_{1j} + \beta_3 z_{3j} + u_i(z_{1j} + \lambda_3 z_{3j}) \\ &= \beta_j + u_i\lambda_j \end{aligned} \tag{6.3}$$

where $z_{1j} = 1$ if $j = 1$ or $j = 2$ and 0 otherwise and $z_{3j} = 1$ if $j = 3$ and 0 otherwise. Note that we are constraining the factor loadings of responses 1 and 2 to be equal by simply using a single dummy variable z_{1j} equal to the sum of the individual dummy variables for responses 1 and 2.

There may be other covariates not assumed to be subject to measurement error. We can add another covariate, x_i, to the disease model

$$\eta_{i3} = \text{logit}(\pi_{i3}) = \beta_3 + \beta_4 x_i + u_i \lambda_3 \tag{6.4}$$

thus assuming a direct effect of the covariate on the risk of disease. This model is shown as a path diagram below:



(In the diagram, circles represent latent variables and rectangles observed variables. Arrows between variables represent linear relations and the little arrows pointing to the rectangles represent residual errors, or in the case of y_3, the binomial variabilty.)

However, it may be that the covariate has an indirect effect on disease by affecting the exposure:

$$u_i = \gamma x_i + \zeta_i \tag{6.5}$$

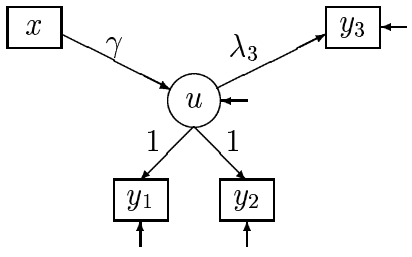
where ζ_i is a residual error term. The measurement model now is

$$\eta_{ij} = \beta_j + \gamma x_i + \zeta_i \tag{6.6}$$

and the disease model is

$$\eta_{i3} = \beta_3 + \gamma \lambda_3 x_i + \zeta_i \lambda_3 \tag{6.7}$$

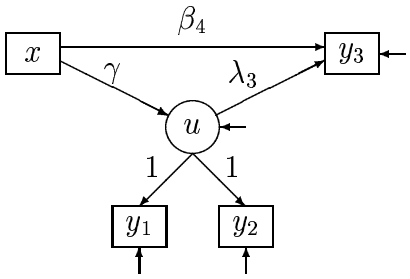
The coefficient of x_i (representing the indirect effect of x_i on the risk of disease), $\gamma\lambda_3$, in the disease model is the product of the coefficient of x_i in the measurement model and the log odds ratio λ_3 - this represents a nonlinear constraint for the parameters. In `gllamm`, this model can therefore only be estimated by specifying the regression of u_i on x_i in (6.5) directly. The model is shown as a path diagram below:



If there are both direct and indirect effects of x on the risk, the disease model becomes

$$\eta_{i3} = \beta_3 + (\beta_4 + \gamma\lambda_3)x_i + \zeta_i\lambda_3 \tag{6.8}$$

In this model, we can estimate the coefficient of x freely, giving an estimate of $\beta_4 + \gamma\lambda_3$ with its standard error. Alternatively, we can explicity specify the regression in (6.5), to obtain estimates of all the individual parameters and their standard errors. This model is shown as a path diagram below:



6.1.1 Data preparation

The data are in *diet.dta*. The variable `r` contains the logarithm of the dietary fibre measurements and `chd` is the binary disease indicator. Those subjects who had two fibre measurements have two lines of data; the variable `t` indicates whether `r` corresponds to the first measurement of fibre (`t`=1) or the second measurement (`t`=2). The men had two types of occupation; `occ`=1: bus staff (drivers and conductors) and `occ`=0: bank staff.

```
. use diet, clear
. sort id t
. list id t r chd occ in 210/220
```

	id	t	r	chd	occ
210.	214	1	2.85	0	0
211.	215	1	2.76	0	0
212.	216	1	2.59	0	0
213.	217	1	3.06	0	0
214.	218	1	3.14	0	0
215.	219	1	2.75	0	0
216.	219	2	2.7	0	0
217.	220	1	2.6	0	0
218.	220	2	2.69	0	0
219.	221	1	2.77	0	0
220.	221	2	2.85	0	0
221.	222	1	2.31	0	0
222.	222	2	2.53	0	0
223.	223	1	2.98	0	0
224.	223	2	3.29	0	0
225.	224	1	2.44	1	0

Note that, for instance, subject 214 had only one measurement of fibre whereas subject 219 had two. We need to stack the variables `r` and `chd` into a single response variable, `resp` and create two dummy variables, `diet` for the fibre measurements and `chd` for disease status (CHD). We will use the `reshape` command but first we must replace one value of `chd` by a missing value for those subjects who have two lines of data:

```
replace chd=. if t==2
rename r resp1
rename chd resp2
gen n=_n
reshape long resp, i(n) j(var)
drop if resp==.
drop n
sort id t var
tab var, gen(i)
rename i1 diet
rename i2 chd
```

The variable `resp` now contains the responses for CHD and log fibre and the variables `diet` and `chd` indicate whether the observation in `resp` is log fibre or whether it is CHD status, respectively. The variable `var` is 1 for diet measurements and 2 for CHD measurements.

We will include occupation as a covariate in the model in the three ways outlined in the previous subsection. To do this, we must create interactions between `occ` and the dummy variables `diet` and `chd`:

```
gen occd=occ*diet
gen occc=occ*chd
```

The data now look like this:

```
. sort id var t
. list id resp diet var occc occd  in 419/431, nolab

            id       resp       diet        var       occc       occd
419.       214       2.85          1          1          0          0
420.       214          0          0          2          0          0
421.       215       2.76          1          1          0          0
422.       215          0          0          2          0          0
423.       216       2.59          1          1          0          0
424.       216          0          0          2          0          0
425.       217       3.06          1          1          0          0
426.       217          0          0          2          0          0
427.       218       3.14          1          1          0          0
428.       218          0          0          2          0          0
429.       219       2.75          1          1          0          0
430.       219        2.7          1          1          0          0
431.       219          0          0          2          0          0
```

Note that unit 219 has two responses for diet. We have a single dummy variable for diet, the sum of dummy variables for the first and second measurements of diet, instead of separate dummy variables for the measurements of diet. Using the single variable for diet in specifying model implies that the two diet measurements are treated as interchangeable. Using such sums of dummy variables is a convenient way to impose equality constraints.

6.1.2 Parameter estimation

We now specify a factor model where u_i is the factor with mixed responses, continuous fibre intake and dichotomous CHD status. We must specify an equation to define the variables whose linear combination ($\boldsymbol{\lambda}'\mathbf{z}$ in equation (1.2)) multiplies the latent variable, here the dummy variables `diet` and `chd`. By specifying `diet` first, we ensure that the loading for `diet` (in the measurement model) is set to 1.

```
. eq id: diet chd
```

Direct and indirect effects of occupation on CHD

We can estimate the model with direct and indirect effects of occupation on CHD by including occupation in the measurement and disease models. We will specify an identity link for the responses that are fibre measurements (`var`=1) and a logit link for the heart disease responses (`var`=2). This is done by simply listing both links in the `link()` option and specifying the 'key' to which link applies to which observation, i.e. `var`, in the `lv()` option. Similarly, we specify two families in the `family()` option and specify `var` as the key to which family applies to which observation in the `fv()` option.

```
. gllamm resp diet chd occc occd, /*
>         */ nocons i(id) eqs(id) link(ident logit) /*
>         */ family(gauss binom) lv(var) fv(var) /*
>         */ nip(20) trace

General model information
------------------------------------------------------------------------------
dependent variable:         resp
family:                      gauss binom
link:                        ident logit
denominator:                1
equation for fixed effects  diet chd occc occd

Random effects information for 2 level model
------------------------------------------------------------------------------
***level 1 equation:

   log standard deviation
   lns1: _cons


***level 2 (id) equation(s):
   (1 random effect(s))


   lambdas for random effect 1
   id1l: chd
   standard deviation for random effect 1
   id1 : diet

>>>>> iteration log omitted
number of level 1 units = 742
number of level 2 units = 333

Condition Number = 57.294908

gllamm model

log likelihood = -186.8994

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |   2.863836   .0236992   120.84   0.000     2.817387    2.910286
         chd |  -1.977043   .2569675    -7.69   0.000    -2.480689   -1.473396
```

```
        occc |   .0478208   .3327215     0.14   0.886    -.6043012    .6999429
        occd |  -.1209673   .0326426    -3.71   0.000    -.1849455   -.0569891
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  .02161557 (.00354728)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .0701073 (.00737485)

    loadings for random effect 1
    chd: -1.9532186 (.72506298)

------------------------------------------------------------------------------
```

The measurement error variance of log-fibre is 0.02 and the residual variance of latent exposure is 0.07. The odds ratio of CHD for unit increase in true fibre intake is given by

```
. disp exp(-1.9532186)
.14181688
```

The effect of occupation on diet is estimated as -0.12 with bus staff eating less fibre than bank staff. While this implies that bank staff should be less at risk of CHD than bus staff, the estimate of the total (direct and indirect) effect of fibre on heart disease, $\beta_4 + \gamma\lambda_3$, is positive but not significant (estimate=0.05, se=0.33).

The model has the structure of a single factor model with covariates and is theoretically identified. However, the condition number is 57.3. The standard errors do not look very large but we could check if there are large correlations between the parameter estimates:

```
. matrix v=e(V)
. matrix c=corr(v)
. matrix list c

symmetric c[7,7]
                resp:       resp:       resp:       resp:       lns1:
                diet         chd        occc        occd       _cons
 resp:diet         1
  resp:chd  -.1482546           1
 resp:occc   .11481182 -.71974881           1
 resp:occd  -.72846579  .10799854 -.15278686           1
lns1:_cons  -.02318717 -.01645563  .00204998  .01688886           1
  id1l:chd   .00343443  .26874086 -.01229009 -.00250121 -.11637887
  id1:diet   .01915976  .00882363 -.00271535 -.01395543 -.40922513

                id1l:       id1:
                 chd        diet
  id1l:chd          1
  id1:diet  .09674343           1
```

The coefficients of `occc` and `chd` are quite highly negatively correlated as are the coefficients of `occd` and `diet`. After a bit of experimentation, we found that the condition number decreases to

15.6 if the fibre measurements are multiplied by 2. The parameter estimates change as expected (e.g. the log odds and its standard error halve), confirming that the paramter estimates can be trusted.

We can fit the same model but estimate the parameter β_4 directly by specifying the regression of u_i on occupation using the `geqs()` option. We first define an equation for this regression:

```
eq f1: occ
```

The second character of the equation name must be a number to indicate which latent variable is to be regressed on the covariates on the right hand side of the equation. Here there is only one latent variable and the second character must be a '1'. We no longer should include occuption in the measurement model:

```
. gllamm resp diet chd occc, /*
>         */ nocons i(id) eqs(id) link(ident logit) /*
>         */ family(gauss binom) lv(var) fv(var) /*
>         */ nip(20) geqs(f1) trace

General model information
------------------------------------------------------------------------------
dependent variable:         resp
family:                      gauss binom
link:                        ident logit
denominator:                1
equation for fixed effects  diet chd occc

Random effects information for 2 level model
------------------------------------------------------------------------------
***level 1 equation:

   log standard deviation
   lns1: _cons


***level 2 (id) equation(s):
   (1 random effect(s))


   lambdas for random effect 1
   id1l: chd
   standard deviation for random effect 1
   id1 : diet

Regressions of random effects on covariates:
   equation for random effect 1
   f1: occ

>>>> iteration log omitted

number of level 1 units = 742
number of level 2 units = 333

Condition Number = 57.667102

gllamm model
```

```
log likelihood = -186.8994

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |   2.863836   .0236992   120.84   0.000     2.817387    2.910286
         chd |  -1.977042   .2569674    -7.69   0.000    -2.480689   -1.473396
        occc |  -.1884548   .3395229    -0.56   0.579    -.8539075     .476998
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  .02161557 (.00354728)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .0701073 (.00737485)

    loadings for random effect 1
    chd: -1.9532186 (.72506297)


Regressions of latent variables on covariates
------------------------------------------------------------------------------

    random effect 1 has 1 covariates:
    occ: -.12096728 (.03264255)
------------------------------------------------------------------------------
```

The estimate of the direct effect of occupation on diet $\hat{\beta}_4$ is negative though not significant. For the same fibre intake, bus staff are at reduced risk of heart disease; combined with the increased risk due to lower fibre intake, the total effect of occupation on diet $(\beta_4 + \gamma\lambda_3)$ is negligible as we saw using the previous parameterisation.

Direct effect of diet on heart disease

By omitting the `geqs()` option, we can estimate the model with no effect of occupation on fibre intake. Starting from the previous parameter estimates, we can use the `skip` option to drop this term.

```
. matrix a=e(b)
. gllamm resp diet chd occc, /*
>         */ nocons i(id) eqs(id) link(ident logit) /*
>         */ family(gauss binom) lv(var) fv(var) /*
>         */ nip(20) from(a) skip trace

>>> output omitted

number of level 1 units = 742
number of level 2 units = 333

Condition Number = 55.683325

gllamm model
```

```
log likelihood = -193.61697

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |   2.799631    .016715   167.49   0.000     2.766871    2.832392
         chd |  -1.874724   .2481894    -7.55   0.000    -2.361166   -1.388281
        occc |  -.1404696   .3353402    -0.42   0.675    -.7977243    .5167852
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  .02196796 (.003558)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .07330515 (.00770257)

    loadings for random effect 1
    chd: -1.9479411 (.72095781)

------------------------------------------------------------------------------
```

As before, there is no significant direct effect of occupation on CHD.

Indirect effect of diet on heart disease

We now omit the direct effect of occupation of heart disease and retain the effect of occupation on fibre intake.

```
. gllamm resp diet chd, /*
>         */ nocons i(id) eqs(id) link(ident logit) /*
>         */ family(gauss binom) lv(var) fv(var) /*
>         */ nip(20) from(a) geqs(f1) skip trace

>>>> output omitted

number of level 1 units = 742
number of level 2 units = 333

Condition Number = 55.853179

gllamm model

log likelihood = -187.05329

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |    2.86354   .0236972   120.84   0.000     2.817095    2.909986
         chd |  -2.068451   .2012525   -10.28   0.000    -2.462898   -1.674003
------------------------------------------------------------------------------

Variance at level 1
```

```
------------------------------------------------------------------------------
  .02156935 (.0035304)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .07018241 (.00736356)

    loadings for random effect 1
    chd: -1.8582014 (.7012236)


Regressions of latent variables on covariates
------------------------------------------------------------------------------

    random effect 1 has 1 covariates:
    occ: -.1203111 (.03263076)
------------------------------------------------------------------------------
```

A semi-parametric mixture model may also be fitted to this dataset using `gllamm` (see Rabe-Hesketh and Pickles (2001)).

Chapter 7

Continuous time to event or survival data

7.1 Proportional hazards models for multiple event data

We assume that, conditional on the random effects, the hazards of any two units are proportional and can be modelled as

$$h_{ij}(t) = h^0(t)\exp(\eta_{ij}) \tag{7.1}$$

where t is time, $h^0(t)$ is the 'baseline' hazard and η_{ij} is the linear predictor of GLLAMMs. Here we have used two subscripts, i for level 2 units (e.g. subjects) and j for level one units (e.g. occasions), but higher level models can be defined in the same way. For two level models, the linear predictor will typically have the form

$$\eta_{ij} = \boldsymbol{\beta}'\mathbf{x}_{ij} + \mathbf{u}_i^{(2)\prime}\mathbf{z}_{ij}^{(2)} \tag{7.2}$$

although factor models can be useful for structuring the covariance matrix in multivariate survival problems.

We will now consider the likelihood conditional on the random effects, i.e. we will ignore that there are random effects in the linear predictor. If a level 1 unit ij was observed from time t_0 and failed or was censored at time t, where δ_{ij} is 1 if the unit failed and 0 otherwise, the unit's contribution to the likelihood is

$$l_{ij} = h_{ij}(t)^{\delta_{ij}}\exp(-\int_{t_0}^{t} h_{ij}(T)dT) \tag{7.3}$$

A piecewise exponential model assumes that the (conditional) baseline hazard function is piecewise constant, with $h^0(T) = h_s$ for $t_{s-1} \leq T < t_s$, $s = 1, 2, \ldots S$ and interval lengths $y_s = t_s - t_{s-1}$. Let $\theta_{ij} = \exp(\eta_{ij})$. Clayton (1988) shows that for a unit that was censored or failed in the kth interval, the unit's contribution to the likelihood (we are again ignoring the random effects) becomes

$$l_{ij} = (h_k\theta_{ij})^{\delta_{ij}}\exp(-\sum_{s=1}^{k} h_s\theta_{ij}y_s) \tag{7.4}$$

and this can be rewritten as

$$l_{ij} = \prod_{j=1}^{k}(h_s\theta_{ij})^{d_{ijs}}\exp(-h_s\theta_{ij}y_s) \tag{7.5}$$

where $d_{ijs} = 0$ for $s < k$ and $d_{ijk} = \delta_{ij}$. This is proportional to the contribution to the likelihood of k (conditionally) independent Poisson processes with means $h_s\theta_{ij}y_s$. Therefore, by representing

each unit by a number of observations (or 'risk sets') equal to the number of time intervals preceding that unit's failure (or censoring) time, the model may be fitted by Poisson regression using d_{ijs} as the dependent variable, $\log(y_s)$ as an offset and dummies for the time intervals as explanatory variables.

Therefore, one approach to multilevel survival modelling is to divide the follow-up period into intervals over which the hazard can be assumed to be constant and use Poisson regression with random effects. Another approach is to define as many intervals as there are unique failure times with each interval starting at (just after) a unique failure time and ending at (just after) the next largest unique failure time. The units contributing to likelihood for given intervals then correspond to the 'risk sets' of Cox's propportional hazards model and no assumption of piecewise constant hazards is made. The Poisson model with a separate constant for each intervals or risk sets yields identical estimates to the Cox's proportional hazards model. However, we will model the baseline hazard function as a smooth function of time. The data manipulation for these methods is very easy using Stata's `stsplit` command (available in Stata 7).

7.2 Proportional hazards model with random coefficients

We will analyse the dataset published in Danahy *et al.* (**?**) and previously analysed by Pickles and Crouchley (1995; 1994). Here subjects with coronary heart disease participated in a randomised crossover trial comparing Isorbide dinitrate (ISDN) with placebo. Before receiving the drug (or placebo), subjects were asked to exercise on exercise bikes to the onset of angina pectoris or, if angina did not occur, to exhaustion. The exercise time and outcome (angina or exhaustion) were recorded. The drug (or placebo) was then taken orally and the exercise test was repeated one hour, three and five hours after drug (or placebo) administration. We therefore have repeated "survival" times per subject pre and post administration of both an active drug and a placebo.

Each subject repeated the experiment 4 times with a placebo and 4 times with ISDN. There are therefore times to angina or exhaustion, t_{icj} for occasions j, condition c (drug versus placebo) within subjects i. The variable d_{icj} is 1 if angina occurred and 0 otherwise. (We do not know the order in which placebo and ISDN were given since this was not reported in the original paper).

Since the subjects started each of the eight experiments at rest, so that the same processes leading to angina or exhaustion can be assumed to begin at the start of each experiment, we will assume that the hazard functions for the experiments are proportional. We will therefore define the time scale as starting at 0 at the beginning of each experiment. This proportionality assumption allows us express the treatment effect as a *hazard ratio*. This is achieved by introducing a covariate x_T equal to 1 after administration of the drug and equal to 0 otherwise, i.e.

$$x_T = \begin{cases} 1 \text{ if } j = 2,3,4 \\ 0 \text{ otherwise} \end{cases} \tag{7.6}$$

In addition, we will allow for a linear decline in the drug effect using another covariate x_D

$$x_D = \begin{cases} j-3 \text{ if } j = 2,3,4 \\ 0 \text{ otherwise} \end{cases} \tag{7.7}$$

The form of the data is illustrated in the Table below:

occasion j	Condition $c=1$ Placebo t_{i1j}	Condition $c=2$ ISDN t_{i2j}	x_T	x_D
1	t_{i11}	t_{i21}	0	0
2	t_{i12}	t_{i22}	1	-1
3	t_{i13}	t_{i23}	1	0
4	t_{i14}	t_{i24}	1	1

Each subject repeated the four experiments in the placebo condition where there was no censoring. We can include the time to Angina in the placebo condition t_{i0j} as a covariate in the model for the log hazard in the treatment condition at time t_{i1j}. Including a random intercept as well as random treatment effects, the model is

$$h_{ij}(t) = h^0(t)\exp(\eta_{ij}) \tag{7.8}$$

$$\ln(h^0(t)) = \alpha_0 + \alpha_1 t_{i1j} + \alpha_2 t_{i1j}^2 + \cdots \tag{7.9}$$

$$\eta_{ij} = u_{0i} + \beta_1 t_{i0j} + (\beta_2 + u_{1i})x_{Ti1j} + \beta_3 x_{Di1j} \tag{7.10}$$

where (u_{0i}, u_{1i}) are assumed to have a bivariate normal distribution. Note that adjusting for the baseline survival times is likely to reduce the random intercept variance.

The linear predictor in `gllamm` will include all the terms for the log baseline hazard in equation (7.9) as well as the terms in equation (7.10).

7.2.1 Data preparation

First we read the data and list the first twelve observations:

```
. use angina4, clear
. list subj occ secondp secondi unceni in 1/12

        subj        occ   secondp    secondi     unceni
 1.        1          1       150        136          1
 2.        1          2       172        445          0
 3.        1          3       118        393          0
 4.        1          4       143        226          1
 5.        2          1       205        250          1
 6.        2          2       287        306          1
 7.        2          3       211        206          1
 8.        2          4       207        224          1
 9.        3          1       221        215          1
10.        3          2       244        232          1
11.        3          3       147        258          1
12.        3          4       250        268          1
```

The data are already in long form with each subject's four time measurements (in seconds) under the two conditions stored in `secondp` for the placebo condition and in `secondi` for the ISDN condition. The variable `unceni` is 1 when the `secondi` refers to the time to angina 0 when `secondi` refers to the time to censoring. The subject and occasion indices are `subj` and `occ`.

We will now construct the necessary covariates after keeping only those variables we need in the data:

```
. keep subj occ secondp secondi unceni
. gen after=occ>1
```

```
. gen decl=cond(occ>1,occ-3,0)
. sort subj occ
. gen id=_n
. list id subj occ after decl in 1/12

           id       subj        occ      after       decl
  1.        1          1          1          0          0
  2.        2          1          2          1         -1
  3.        3          1          3          1          0
  4.        4          1          4          1          1
  5.        5          2          1          0          0
  6.        6          2          2          1         -1
  7.        7          2          3          1          0
  8.        8          2          4          1          1
  9.        9          3          1          0          0
 10.       10          3          2          1         -1
 11.       11          3          3          1          0
 12.       12          3          4          1          1
```

The coefficient of `after` will represent the treatment effect overall (post-treatment minus baseline) and that of `decl` will represent the linear change in treatment effect over the three post-treatment conditions. The variable `id` labels all combinations of subjects and occasions.

We will also standardise `secondp` which will be used as a covariate:

```
egen timep = std(secondp)
replace secondp=timep
drop timep
```

First we analyse the data using Stata's Cox regression procedure so that we can check the correctness of the expansion of the data to risk sets later on.

```
. stset secondi, failure(unceni) id(id)

                id:  id
     failure event:  unceni ~= 0 & unceni ~= .
obs. time interval:  (secondi[_n-1], secondi]
 exit on or before:  failure

------------------------------------------------------------------------------
       84  total obs.
        0  exclusions
------------------------------------------------------------------------------
       84  obs. remaining, representing
       84  subjects
       71  failures in single failure-per-subject data
    27066  total analysis time at risk, at risk from t =         0
                             earliest observed entry t =         0
                                  last observed exit t =       743

. stcox secondp after decl

         failure _d:  unceni
   analysis time _t:  secondi
                 id:  id

Iteration 0:   log likelihood = -261.50926
```

```
Iteration 1:   log likelihood = -232.34921
Iteration 2:   log likelihood = -229.30664
Iteration 3:   log likelihood = -229.23612
Iteration 4:   log likelihood = -229.23608
Refining estimates:
Iteration 0:   log likelihood = -229.23608

Cox regression -- Breslow method for ties

No. of subjects =           84                     Number of obs   =        84
No. of failures =           71
Time at risk    =        27066
                                                   LR chi2(3)      =     64.55
Log likelihood  =   -229.23608                     Prob > chi2     =    0.0000

------------------------------------------------------------------------------
          _t |
          _d | Haz. Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     secondp |   .3100321   .0604764    -6.00   0.000     .2115277    .4544082
       after |   .3480569   .1000469    -3.67   0.000     .1981424    .6113968
        decl |   1.873714    .367028     3.21   0.001     1.276344    2.750672
------------------------------------------------------------------------------
```

We now expand the dataset to risk sets. For each unique failure time (or risk set), there will be a record for each `id` with a failure time or censoring time greater than that failure time. Having used `stset` above to specify the survival time structure of the data, we can use Stata's `stsplit` command to achieve this:

```
. stsplit, at(failures) riskset(RS)
(63 failure times)
(2847 observations (episodes) created)

. sort RS id

. list RS id secondi unceni in 2901/2911

             RS         id    secondi     unceni
2901.        60         22        580          1
2902.        60         23        580          .
2903.        60         58        580          .
2904.        60         62        580          .
2905.        60         74        580          .
2906.        60         75        580          .
2907.        61         23        613          1
2908.        61         58        613          .
2909.        61         62        613          .
2910.        61         74        613          .
2911.        61         75        613          .
```

There are 63 risk-sets. The risk-sets are labelled in increasing order of the associated survival times and therefore in decreasing order of risk set size (as fewer individuals 'outlive' survive beyond the times). The censoring indicator has been changed to missing for censored observations. We will change these missing values to 0 since this will be our dependent variable in the Poisson regression.

```
replace unceni = 0 if unceni == .
```

We can verify that the data are correct by rerunning the Cox regression (using the same command as before) which yields identical results.

Now we need to compute the lengths of the intervals between unique failure times so that we can use the log interval lengths as an offset in the Poisson regression:

```
. sort id secondi
. by id: gen y=cond(_n>1,secondi-secondi[_n-1],secondi)
. gen lny = ln(y)
. list RS id secondi y lny in 2901/2911

             RS         id    secondi          y        lny
2901.        28         84        231          1          0
2902.        29         84        232          1          0
2903.        30         84        235          3   1.098612
2904.        31         84        248         13   2.564949
2905.        32         84        250          2   .6931472
2906.        33         84        258          8   2.079442
2907.        34         84        264          6   1.791759
2908.        35         84        265          1          0
2909.        36         84        268          3   1.098612
2910.        37         84        280         12   2.484907
2911.        38         84        290         10   2.302585
```

We can check that this is correct by comparing the result of fitting an exponential regression model using

```
streg secondp after decl, dist(exp)
```

with that of running a simple Poisson regression using

```
poisson unceni secondp after decl, offset(lny) irr
```

(both models assume constant hazards over the entire period which is unrealistic, but here we just want to check our data manipulation). Both models give the same result confirming that our data has been set up correctly.

7.2.2 Parameter estimation

Assuming a constant baseline hazard would correspond to the exponential model. Allowing the baseline hazard to vary freely between risk sets corresponds to Cox's regression model (We would get identical results as Cox's regression by using fixed effects Poisson, i.e. `xtpois unceni secondp after decl, i(RS) fe offset(lny) irr` or `xi: poisson unceni secondp after decl i.RS, offset(lny) irr`).

We will model the log baseline hazard as a smooth function of time by using polynomial terms. One way of assessing that the model for the baseline hazard is sufficiently flexible, is to compare the estimates of the effects of interest with those of Cox's regreession model.

Orthogonal polynomials can be created using `orthpoly`:

```
orthpoly secondi, gen(t1-t4) degree(4)
```

and included in the Poisson regression:

```
. poisson unceni t1-t4 secondp after decl, offset(lny) irr nolog

Poisson regression                                Number of obs   =       2931
                                                  LR chi2(7)      =      96.93
                                                  Prob > chi2     =     0.0000
Log likelihood = -328.31446                       Pseudo R2       =     0.1286

------------------------------------------------------------------------------
      unceni |        IRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   2.073082   .2668426     5.66   0.000     1.610838    2.667973
          t2 |   .6969082   .0744861    -3.38   0.001     .5651953    .8593154
          t3 |   1.488465   .1337025     4.43   0.000     1.248184       1.775
          t4 |   .7781663   .0564711    -3.46   0.001     .6749961    .8971057
     secondp |     .34998   .0642334    -5.72   0.000     .2442409    .5014967
       after |   .3731128   .1043298    -3.53   0.000     .2156886    .6454358
        decl |   1.746769   .3315779     2.94   0.003     1.204086    2.534041
         lny |   (offset)
------------------------------------------------------------------------------
```

The fixed effects parameters are quite close to those of the Cox model. We could also model the log baseline hazard using fractional polynomials or splines (see help for `fracpoly` and `mkspline`).

We can now fit a simple random intercept model in `gllamm` by using the `offset()` option, specifying the Poisson family (the log link is the default link for Poisson) and including a random intercept in the linear predictor:

```
. gen cons=1
. eq cons: cons
. gllamm unceni t1-t4 secondp after decl, i(subj) nip(8) /*
>   */ eqs(cons) family(poiss) offset(lny) trace eform

General model information
------------------------------------------------------------------------------
dependent variable:         unceni
family:                     poiss
link:                       log
offset:                     lny
equation for fixed effects  t1 t2 t3 t4 secondp after decl _cons

Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (subj) equation(s):
   (1 random effect(s))

   standard deviation for random effect 1
   subj1 : cons

>>> output omitted

number of level 1 units = 2931
number of level 2 units = 21

Condition Number = 10.960424

gllamm model
```

```
log likelihood = -314.13229

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   4.945214   1.104327     7.16   0.000     3.192283    7.660707
          t2 |   .5294363   .0679726    -4.95   0.000     .4116526    .6809209
          t3 |   1.515796   .1536843     4.10   0.000     1.242621    1.849025
          t4 |   .7701484   .0622239    -3.23   0.001     .6573578    .9022919
     secondp |   .2767016   .0661491    -5.37   0.000     .1731892    .4420816
       after |   .3783459   .1112643    -3.31   0.001     .2126018    .6733039
        decl |   2.308358   .4780472     4.04   0.000     1.538241    3.464033
         lny |   (offset)
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (subj)

    var(1): 3.0482593 (1.2907518)
------------------------------------------------------------------------------
```

Before interpreting the results, we will check whether they can be relied on by increasing the number of quadrature points. Estimating the model again with 20 quadrature points gives a much lower estimate of the random intercept variance:

```
. matrix a=e(b)
. gllamm unceni t1-t4 secondp after decl, i(subj) nip(20) /*
>   */ eqs(cons) family(poiss) from(a) offset(lny) eform
number of level 1 units = 2931
number of level 2 units = 21

Condition Number = 9.4673366

gllamm model

log likelihood = -314.95715

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |    4.55074   1.106933     6.23   0.000     2.825099    7.330446
          t2 |   .5217291   .0785237    -4.32   0.000     .3884485    .7007395
          t3 |     1.5743    .172777     4.14   0.000     1.269607    1.952115
          t4 |   .7721425   .0611082    -3.27   0.001     .6611994    .9017007
     secondp |   .2313422   .0669501    -5.06   0.000     .1311953    .4079352
       after |   .4107819   .1253004    -2.92   0.004      .225928    .7468829
        decl |   2.207636   .4625463     3.78   0.000     1.464141    3.328681
         lny |   (offset)
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------
```

```
***level 2 (subj)

    var(1): 1.6963166 (.78712772)
------------------------------------------------------------------------------
```

but the estimates seem to stabilise, the 30 point estimates being:

```
. matrix a=e(b)
. gllamm unceni t1-t4 secondp after decl, i(subj) nip(30) /*
>   */ eqs(cons) family(poiss) from(a) offset(lny) eform
number of level 1 units = 2931
number of level 2 units = 21

Condition Number = 9.419333

gllamm model

log likelihood = -315.00414

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   4.493769   1.106889     6.10   0.000     2.772974    7.282419
          t2 |    .522476   .0813625    -4.17   0.000     .3850464    .7089565
          t3 |   1.575571   .1737476     4.12   0.000      1.26932    1.955713
          t4 |   .7716742   .0609903    -3.28   0.001     .6609341     .900969
     secondp |   .2343819   .0732762    -4.64   0.000     .1270012    .4325538
       after |   .4108853   .1262044    -2.90   0.004     .2250463    .7501866
        decl |   2.200337   .4634823     3.74   0.000     1.456099    3.324969
         lny |   (offset)
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (subj)

    var(1): 1.6563003 (.86840339)
------------------------------------------------------------------------------
```

The regression coefficients have been exponentiated in the output as indicated by `exp(b)` because we used the `eform` option. The parameters can therefore be interpreted as conditional hazard ratios (conditional on the random effects). The estimates suggest that ISDN reduces the hazards compared with baseline. and that there is linear decline in this treatment effect. The coefficient of `after` compares occasions 3 and 1 since `decl`=0 for both occasions. To compare occasions 1 and 2, we could use `lincom`:

```
lincom after + decl, eform
```

There is a large random intercept variance. Note that the exponential of the random intercept is usually referred to as a *frailty*, so our variance estimate is not the frailty variance. Comparing the log-likelihood with that of the ordinary Poisson model (see previous section), indicates that the intercept varies significantly between subjects:

```
. disp chiprob(1, 2*(328.31446 -315.00414))
2.476e-07
```

Note that these models could have been fitted much more quickly using the `xtpois` command:

```
xtpois unceni t1-t4 secondp after decl, i(subj) quad(30) offset(lny) normal irr
```

giving exactly the same results. With the `xtpois` command, we can also assume a gamma distribution of the frailty by omitting the `normal` and `quad()` options above. This gives very similar fixed effects estimates which is reassuring.

We now allow the treatment effect to vary randomly between subjects by introducing a random coefficient for `after` (This cannot be done in `xtpois`):

```
. gllamm unceni t1-t4 secondp after decl, i(subj) nrf(2) eqs(cons after) /*
>    */ family(poiss) offset(lny) nip(8) trace eform

General model information
------------------------------------------------------------------------------
dependent variable:         unceni
family:                     poiss
link:                       log
offset:                     lny
equation for fixed effects  t1 t2 t3 t4 secondp after decl _cons

Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (subj) equation(s):
   (2 random effect(s))


   diagonal element of cholesky decomp. of covariance matrix
   subj1 : cons

   diagonal element of cholesky decomp. of covariance matrix
   subj2 : after

   off-diagonal elements
   subj2_1: _cons

>>>> output omitted

number of level 1 units = 2931
number of level 2 units = 21

Condition Number = 35.401002

gllamm model

log likelihood = -307.71846

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   6.201066    1.90094     5.95   0.000     3.400401    11.30844
```

```
          t2 |   .4694182   .0894685    -3.97   0.000     .3230915    .6820156
          t3 |   1.664008   .2161324     3.92   0.000     1.290018    2.146422
          t4 |    .749202   .0649883    -3.33   0.001     .6320666     .888045
     secondp |   .1844515   .0862036    -3.62   0.000     .0738027    .4609909
       after |   .2884662   .1261288    -2.84   0.004     .1224379    .6796321
        decl |   2.271276    .499788     3.73   0.000     1.475589    3.496025
         lny |   (offset)
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (subj)

    var(1): .48137109 (.5182014)
    cov(1,2): .74458779 (.84153151) cor(1,2): .74346509

    var(2): 2.0836783 (1.4310411)
------------------------------------------------------------------------------
```

To assess how good the 8 point quadrature solution is, we will run the model with 20 points - this will take about $20\times20/(8\times8) = 6.25$ times as long!

```
. matrix a=e(b)
. gllamm unceni t1-t4 secondp after decl, i(subj) nrf(2) eqs(cons after) /*
>   */ family(poiss) offset(lny) from(a) nip(20) eform

number of level 1 units = 2931
number of level 2 units = 21

Condition Number = 19.272222

gllamm model

log likelihood = -307.79395

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   5.819054   1.627473     6.30   0.000     3.363477    10.06738
          t2 |   .4772048   .0856703    -4.12   0.000     .3356547    .6784486
          t3 |   1.660404   .2044309     4.12   0.000     1.304407    2.113561
          t4 |   .7514037   .0647278    -3.32   0.001      .634671    .8896066
     secondp |   .1863612   .0626826    -5.00   0.000     .0963949     .360294
       after |   .3114577   .1344426    -2.70   0.007     .1336512    .7258139
        decl |   2.235054   .4859076     3.70   0.000     1.459601    3.422488
         lny |   (offset)
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (subj)

    var(1): .37465037 (.42156288)
    cov(1,2): .72940438 (.42629403) cor(1,2): .90065961
```

```
    var(2): 1.7506091 (1.1129138)
------------------------------------------------------------------------------
```

The estimates of the covariance matrix are quite different but the fixed effects estimates have not changed substantially. We can check how much the log-likelihood for these parameter estimates changes when more quadrature points are used using the `eval` option:

```
. gllamm unceni t1-t4 secondp after decl, i(subj) nrf(2) eqs(cons after) /*
>   */ f(amily poiss) offset(lny) from(a) nip(30) eval
log-likelihood = -307.78548
```

The change in log-likelihood is small.

Using a likelihood ratio test, there is significant variability in the treatment effect (twice the difference in log-likelihoods=14.42). This illustrates how misleading the standard errors of the variance estimates can be. Subjects with a greater random intercept (or frailty) tend to have a larger coefficient of `after`, i.e. a smaller treatment effect. (To test for the significance of this correlation, we would have to run `gllamm` again using the `nocor` option and use another likelihood ratio test).

Chapter 8

Ordinal responses

8.1 Generalising models for ordinal responses

The ordinal models available in `gllamm` can be written as latent response models with

$$y^* = \eta + \epsilon \tag{8.1}$$

where the S observed response categories y_s, $s = 1, \cdots, S$ are generated by applying thresholds κ_s, $s = 1, \cdots, S-1$ to y^* as follows:

$$y = \begin{cases} y_1 & \text{if } y^* \leq \kappa_1 \\ y_2 & \text{if } \kappa_1 < y^* \leq \kappa_2 \\ \vdots & \vdots \\ y_S & \text{if } \kappa_{S-1} < y^* \end{cases} \tag{8.2}$$

where the thresholds κ_s do not vary between subjects.

If the cumulative density function of ϵ is F, the cumulative probability τ_s that the response takes on any value up to and including y_s (conditional on the latent and observed explanatory variables) is

$$\mathrm{P}(y \leq y_s) = F(\kappa_s - \eta), \quad s = 0, \cdots, S \tag{8.3}$$

where $\kappa_0 = -\infty$ and $\kappa_S = \infty$. The probability of the sth response category is then simply

$$\begin{aligned} \mathrm{P}(y = y_s) &= \mathrm{P}(\kappa_{s-1} < y^* \leq \kappa_s) \\ &= F(\kappa_s - \eta) - F(\kappa_{s-1} - \eta). \end{aligned} \tag{8.4}$$

We can equivalently write the model as a cumulative model

$$g(\mathrm{P}(y < y_s)) = \kappa_s - \eta,$$

where $g = F^{-1}$ is the link function.

In `gllamm` F can be the logistic distribution (ordinal logit link, `ologit`), standard normal distribution (ordinal probit link, `opropbit`) or type I extreme value distribution (ordinal complimentary log-log link, `ocll`). If the error term ϵ of the latent response y^* is assumed to have a logistic distribution,

$$\begin{aligned} \Pr(y \leq y_s) &= \Pr(y^* \leq \kappa_s) \\ &= \frac{\exp(\kappa_s - \eta)}{1 + \exp(\kappa_s - \eta)} \end{aligned} \tag{8.5}$$

and we have a *proportional odds* model since the log odds that $y \leq y_s$ (conditional on the latent and observed explanatory variables) are

$$\log\left(\frac{\Pr(y \leq y_s)}{1-\Pr(y \leq y_s)}\right) = \kappa_s - \eta \tag{8.6}$$

so that the odds that the response category is less than or equal to y_s for an individual i is a constant multiple of the odds for another individual i' with odds ratio equal to $\exp(\eta_i - \eta_{i'})$ for all s. This parameterisation is identical to that used in Stata's `ologit` command.

If the error term ϵ of the latent response has a standard normal distribution, we have the probit model with

$$\Pr(y \leq y_s) = \Phi(\kappa_s - \eta) \tag{8.7}$$

where Φ is the cumulative standard normal distribution function. This parameterisation is identical to that used in Stata's `oprobit` command.

If the error term ϵ has an extreme value distribution, this corresponds to an ordinal complimentary log-log link

$$\ln(-\ln(1-\Pr(y \leq y_s))) = \kappa_s - \eta \tag{8.8}$$

or, equivalently,

$$\Pr(y \leq y_s) = 1 - \exp(-\exp(\kappa_s - \eta)) \tag{8.9}$$

Normally the effects of covariates are assumed to be constant across categories s. If F is the logistic distribution, this assumption corresponds to the proportional odds assumption. Using the `thresh()` option, the thresholds can be allowed to depend on covariates x_1 to x_p as

$$\kappa_s = \alpha_{s0} + \alpha_{s1}x_1 + \cdots \alpha_{sp}x_p.$$

Note that the model is not identified if any of the covariates used in the threshold model also appear in the linear predictor η.

If there are several ordinal responses differing in the thresholds and possibly in the number of response categories, we use the same method as for mixed response models. Simply specify the ordinal link, e.g., `ologit` several times in the `link()` option and use `lv()` to specify the variable identifying the responses.

8.2 Ordinal item response models

Item response models for dichotomous items were introduced in Section 4.1. Here we will discuss similar models for ordinal responses. We will analyse six ordinal items relating to delinquency from Udry (1998).

The simplest item response model for ordinal items assumes that the latent response has different means for the different items but the same residual variance $\text{var}(\epsilon)$. Further, the effect of the latent variable u_i is the same for all items and the thresholds κ_s are constant across items j:

$$g(\mathrm{P}(y_{ij} \leq y_s)) = \kappa_s - (\beta_j + u_i), \quad \beta_1 = 0 \tag{8.10}$$

For identification, one of the means (here β_1) must be set to a constant since the thresholds are estimated freely. We can now allow the effect of u_i to differ between items by including factor loadings λ_j in the model

$$g(\mathrm{P}(y_{ij} \leq y_s)) = \kappa_s - (\beta_j + u_i\lambda_j), \quad \beta_1 = 0,\ \lambda_1 = 1 \tag{8.11}$$

Next, we can allow the residual variance to differ between items by using the scaled ordinal probit link, `soprobit`.

$$\mathrm{P}(y_{ij} \leq y_s) = \Phi((\kappa_s - \beta_j - u_i\lambda_j)/\sigma_j), \quad \beta_1 = 0,\ \lambda_1 = 1,\ \sigma_1 = 1 \tag{8.12}$$

where σ_j is the scale, corresponding to the standard deviation of ϵ in the latent response formulation. Such a model was suggested and fitted by Skrondal (1996). Since the thresholds are estimated freely, the scale of one item has to be set to a constant for the model to be identified. The model can be rewritten as

$$\begin{aligned} g(\mathrm{P}(y_{ij} \leq y_s)) &= (\kappa_s - \beta_j)/\sigma_j - u_i\lambda_j/\sigma_j \\ &= \kappa^*_{sj} - u_i\lambda^*_j \end{aligned}$$

where

$$\kappa^*_{sj} = (\kappa_s - \beta_j)/\sigma_j.$$

The model therefore effectively allows a different linear transformation of the thresholds for each item, i.e. the thresholds can be shifted and rescaled for each item.

A more general model allows the thresholds to differ completely between items:

$$g(\mathrm{P}(y_{ij} \leq y_s)) = \kappa_{sj}u_i\lambda_j, \quad \beta_1 = 0 \tag{8.13}$$

Finally, covariates can be incorporated in different ways: (1) the covariate affects the latent response indirectly by affecting the latent variable u_i (2) the covariate has a direct effect on the latent response, possibly in addition to an indirect effect via the latent variable (3) the covariate affects the thresholds. In the last two situations, the effect of the covariate can either be the same for all items or differ between items.

8.3 Three level ordinal logistic regression

A two-level analysis of a subset of the Television School and Family Smoking Prevention and Cessation Project (TVSFP) (Flay *et al.*, 1989) is presented in Hedeker and Gibbons' MIXOR manual (Hedeker and Gibbons, 1996) and also analysed in Hedeker and Gibbons (1994). Schools were randomised to one of four conditions given by different combinations of two factors

TV: a media (television) intervention (1=present, 0=absent)

CC: a social-resistance classroom curriculum (1=present, 0=absent).

One outcome measure is the tobacco and health knowledge scale (THKS) score defined as the number of correct answers to seven items on tobacco and health knowledge. This variable has been collapsed into four ordinal categories.

In addition to the clustering of students in classes, the classes are clustered in schools. First, we will repeat Hedeker and Gibbons' two-level analysis ignoring schools. Then we will fit a three level model incorporating the effect of schools. (The three level model cannot be estimated in the present version of MIXOR but Gibbons and Hedeker (1997) fitted a three level model to the dichotomised THKS score.)

The two-level model can be written as

$$\eta_{ijk} = \beta_0 + \beta_1 x_{Pijk} + \beta_2 x_{Ci} + \beta_3 x_{Ti} + \beta_4 x_{Ci}x_{Ti} + u_j \tag{8.14}$$

where i, j and k denote schools, classes and pupils, respectively, x_{Pijk} is the pre-intervention THKS score, x_{Ci} is a dummy variable for the CC intervention and x_{Ti} is a dummy variable for the TV intervention.

The three level model is

$$\eta_{ijk} = \beta_0 + \beta_1 x_{Pijk} + \beta_2 x_{Ci} + \beta_3 x_{Ti} + \beta_4 x_{Ci} x_{Ti} + u_j^{(2)} + u_{ij}^{(3)} \tag{8.15}$$

where the dependent variable is modelled using a proportional odds model.

8.3.1 Data preparation

The data are available from Hedeker's home page as an ASCII file called *tvsfpors.dat* that is already in the long form. We read the data using `infile` and drop all observations with missing values.

```
infile school class thk a2 const prethk cc tv cctv using tvsfpors.dat, clear
keep school class thk prethk cc tv cctv
for var thk prethk cc tv cctv: summ X
drop if thk==.
drop if prethk==.
drop if cc==.
drop if tv==.
drop if cctv==.
```

Here `thk` and `prethk` are the ordinal THKS score post and pre intervention, respectively and `cc`, `tv` and `cctv` are the dummy variables for the main effects and interaction of the CC and TV interventions.

To speed up estimation, we can now collapse the data and define frequency weights. Since it is unlikely that two classes will have exactly the same number of students and pattern of outcomes and covariates, we will not attempt to form level 2 weights. To form level 1 weights, we need to aggregate data over all groups of children in the same class that have the same outcome. We do not need to worry about the schools and the combination of treatments because these variables are constant within classes. However, we can include these variables in the `by()` option so that they are not dropped from the dataset. The data are therefore collapsed as follows:

```
gen cons=1
collapse (count) wt1=cons ,by(thk prethk cc tv cctv school class)
```

The variables `school`, `class`, `thk`, `cc`, `tv` and `wt1` are listed below for ten observations:

	school	class	thk	cc	tv	wt1
80.	196	196102	4	1	0	2
81.	196	196102	4	1	0	1
82.	196	196102	4	1	0	1
83.	196	196102	2	1	0	1
84.	196	196102	4	1	0	1
85.	197	197101	2	0	0	2
86.	197	197101	1	0	0	1
87.	197	197101	2	0	0	1
88.	197	197101	2	0	0	1
89.	197	197101	4	0	0	1
90.	197	197101	3	0	0	1

8.3.2 Model Fitting

The syntax is identical to that used for an ordinary logistic regression model except that the `ologit` link is specified.

```
. gllamm thk prethk cc tv cctv, i(class) trace link(ologit) family(binom) /*
>   */ weight(wt) nip(10)

General model information
------------------------------------------------------------------------------
dependent variable:          thk
ordinal responses:         ologit
denominator:                  1
equations for fixed effects
                           thk:  prethk cc tv cctv
                           _cut11:  _cons
                           _cut12:  _cons
                           _cut13:  _cons


Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (class) equation(s):

   standard deviation of random effect
   clas: _cons

number of level 1 units = 1600
number of level 2 units = 135

Condition Number = 15.444405

gllamm model

log likelihood = -2115.3811

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
thk          |
      prethk |   .4147969   .0393634    10.54   0.000     .3376461    .4919477
          cc |   .8613649   .1735348     4.96   0.000     .5212429    1.201487
          tv |    .205729   .1706124     1.21   0.228    -.1286653    .5401232
        cctv |  -.3009843    .245123    -1.23   0.219    -.7814166    .1794479
-------------+----------------------------------------------------------------
_cut11       |
       _cons |  -.0757244   .1466208    -0.52   0.606    -.3630959    .2116471
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   1.197719   .1485227     8.06   0.000     .9066203    1.488819
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   2.403267    .157911    15.22   0.000     2.093767    2.712767
------------------------------------------------------------------------------


Variances and covariances of random effects
```

```
------------------------------------------------------------------------------

***level 2 (class)

    var(1): .18882345 (.06388833)
------------------------------------------------------------------------------
```

The output agrees with the results obtained using MIXOR with 10 quadrature points (see MIXOR manual, Hedeker and Gibbons, 1996). As would be expected, the level of knowledge before the intervention is a predictor of the the level of knowledge after the intervention. The social-resistance classroom curriculum has had an effect but the TV intervention has not. The between-class variance is estimated as 0.189. The estimates of the three cut-points κ_1, κ_2 and κ_3 appear at the bottom of the fixed effects table.

We could easily remove the nonsignificant terms from the model by passing the parameter estimates of the full model to `gllamm` as initial parameter estimates using the `from()` and the `skip` options:

```
matrix a=e(b)
gllamm thk prethk cc, i(class) trace link(ologit) family(binom) /*
  */ weight(wt) nip(10) from(a) skip
```

To fit the probit or complimentary log-log model, simply use the oprobit link or indxocll link, respectively, instead of the ologit link.

We now include a random effect for schools by adding `school` to the `i()` option.

```
. gllamm thk prethk cc tv cctv, i(class school) trace link(ologit) /*
>   */ family(binom) weight(wt) nip(10)

General model information
------------------------------------------------------------------------------
dependent variable:          thk
ordinal responses:         ologit
denominator:                  1
equations for fixed effects
                           thk:  prethk cc tv cctv
                           _cut11:  _cons
                           _cut12:  _cons
                           _cut13:  _cons


Random effects information for 3 level model
------------------------------------------------------------------------------


***level 2 (class) equation(s):

   standard deviation of random effect
   clas: _cons

***level 3 (school) equation(s):

   standard deviation of random effect
   scho: _cons

number of level 1 units = 1600
```

```
number of level 2 units = 135
number of level 3 units = 28


Condition Number = 16.636362

gllamm model

log likelihood = -2114.5881

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
thk          |
      prethk |   .4085269   .0396159    10.31   0.000     .3308811    .4861727
          cc |   .8843998   .2098805     4.21   0.000     .4730416    1.295758
          tv |   .2364089   .2048806     1.15   0.249    -.1651497    .6379675
        cctv |  -.3717357   .2958354    -1.26   0.209    -.9515624    .2080909
-------------+----------------------------------------------------------------
_cut11       |
       _cons |  -.0959862   .1689184    -0.57   0.570    -.4270601    .2350878
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   1.177437   .1705139     6.91   0.000     .8432364    1.511639
-------------+----------------------------------------------------------------
_cut13       |
       _cons |    2.38363    .178691    13.34   0.000     2.033402    2.733858
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (class)

    var(1): .14821014 (.06374783)

***level 3 (school)

    var(1): .04487616 (.04254602)
------------------------------------------------------------------------------
```

The variance component for schools is not significant at the 5% level since the log-likelihood changed by less than 1.

We could estimate the model with more quadrature points to make sure that we are evaluating the likelihood sufficiently precisely. Since this is time-consuming, we can simply evaluate the log-likelihood for the 10 point parameter estimates using larger numbers of quadrature points by using the `eval` option:

```
. matrix a= e(b)
. gllamm thk prethk cc tv cctv, i(class school) link(ologit) /*
>  */ family(binom) weight(wt) nip(20) eval from(a)
log-likelihood = -2114.5881
. gllamm thk prethk cc tv cctv, i(class school) link(ologit) /*
>  */ family(binom) weight(wt) nip(30) eval from(a)
log-likelihood = -2114.5881
. gllamm thk prethk cc tv cctv, i(class school) link(ologit) /*
```

```
>  */ family(binom) weight(wt) nip(40) eval from(a)
log-likelihood = -2114.5881
```

The log-likelihood values do not change at all when more quadrature points are used and the 10 point approximation therefore appears to be adequate.

8.4 Item response models with an explanatory variable

8.4.1 Data preparation

The data are read using

```
infile sex y1 y2 y3 y4 y5 y6 using delinq.txt, clear
```

Since many of the response pattern on the 6 items are likely to occur a number of times for each sex, we can collapse the data and construct level 2 weights to make `gllamm` run faster:

```
gen cons=1
collapse (sum) wt2=cons, by(sex y1-y6)
gen id=_n
reshape long y, i(id) j(item)
```

8.4.2 Model fitting

Thresholds constant across items

The model in (8.10) is a simple random intercept model with an `oprobit` link and with non-zero intercepts for items 2 to 6 ($\beta_1 = 0$). The syntax is therefore similar to the `xt` commands. Here we also need to use the `weight()` option since we have level 2 weights in the variable `wt2`.

```
. tab item, gen(i)
. gllamm y i2-i6, i(id) weight(wt) l(oprob) f(binom)

number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 8.1990212

gllamm model

log likelihood = -10226.616

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          i2 |   .1327059   .0368782     3.60   0.000      .060426    .2049857
          i3 |  -.3556413   .0422637    -8.41   0.000    -.4384767   -.2728059
          i4 |  -.3832299   .0426228    -8.99   0.000     -.466769   -.2996908
          i5 |  -.5155239   .0446666   -11.54   0.000    -.6030688   -.4279789
          i6 |  -.0334661   .0386515    -0.87   0.387    -.1092215    .0422894
-------------+----------------------------------------------------------------
_cut11       |
       _cons |    1.98145    .037215    53.24   0.000      1.90851     2.05439
```

```
-------------+------------------------------------------------------------
_cut12       |
       _cons |   2.765931   .0428442    64.56   0.000     2.681957    2.849904
-------------+------------------------------------------------------------
_cut13       |
       _cons |   3.119489   .0467586    66.71   0.000     3.027844    3.211134
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): 1.0110074 (.05052765)
------------------------------------------------------------------------------
```

The estimated constants suggest that the scores for item 2 tend to be higher than those for item 1 whereas the scores for items 3 to 6 tend to be lower than those for item 1.

We can introduce factor loadings using the `eqs()` option (the first loading will automatically be set to one):

```
. eq load: i1-i6
. gllamm y i2-i6, i(id) weight(wt) l(oprob) f(binom) eqs(load)

number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 27.905871

gllamm model

log likelihood = -10157.782

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          i2 |    .047605   .0574449     0.83   0.407    -.0649849    .1601948
          i3 |  -1.095781   .1223369    -8.96   0.000    -1.335557   -.8560054
          i4 |  -.9650579   .1110193    -8.69   0.000    -1.182652    -.747464
          i5 |  -.8419555   .0981992    -8.57   0.000    -1.034422   -.6494887
          i6 |  -.7239922   .1055135    -6.86   0.000    -.9307948   -.5171895
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.695634   .0417319    40.63   0.000     1.613841    1.777427
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.489353   .0460217    54.09   0.000     2.399153    2.579554
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   2.854539   .0493797    57.81   0.000     2.757757    2.951322
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------
```

```
***level 2 (id)

    var(1): .48437902 (.05732434)

    loadings for random effect 1
    i2: 1.1283393 (.08673783)
    i3: 1.9509092 (.15915351)
    i4: 1.7775206 (.1470144)
    i5: 1.4899425 (.12713595)
    i6: 1.9559887 (.15774182)

------------------------------------------------------------------------------
```

A likelihood ratio test shows that this model fits considerably better than the previous:

```
. disp chiprob(5,2*(10226.616 -10157.782))
5.601e-28
```

Finally the `soprobit` (scaled ordinal probit) link can be used together with the `s()` option to allow the scale of the error term in the latent response formulation to vary between items as in (8.13). The `s()` option allows an equation to be specified to introduce level 1 heteroscedasticity when any of the links or densities specified have a scale or standard deviation parameter. (Another application of this option would be to estimate a linear model with a heteroscedastic error term.) The equation definition for the `s()` option should specify the dummy variables for the items since a separate scale parameter is required for each item:

```
. eq het: i1-i6
```

However, we need to constrain the first scale to 1. To use constraints in `gllamm`, we have to find out the equation name and column name for the parameter being constrained as well as the transformation used in `gllamm` to estimate the parameter. We can do this by running `gllamm` with the `noest` and `trace` options to obtain some information on the model without estimating any parameters:

```
. gllamm y i2-i6, i(id) weight(wt) l(soprob) f(binom) eqs(load) s(het) noest trace

General model information
------------------------------------------------------------------------------
dependent variable:          y
ordinal responses:          soprobit
denominator:                 1
equations for fixed effects
                            y:  i2 i3 i4 i5 i6
                            _cut11:  _cons
                            _cut12:  _cons
                            _cut13:  _cons


Random effects information for 2 level model
------------------------------------------------------------------------------
***level 1 equation:

   log standard deviation
   lns1: i1 i2 i3 i4 i5 i6
```

```
***level 2 (id) equation(s):
   (1 random effect(s))


   lambdas for random effect 1
   id1l: i2 i3 i4 i5 i6
   standard deviation for random effect 1
   id1 : i1
```

Under 'level 1 equation', we are informed that the log standard deviation parameters have equation name `lns1` and column names `i1`, `i2`, etc. Constraining the standard deviation to 1 corresponds to setting the log standard deviation to 0. We can now define this contraint (see [R] contraint) and pass it to `gllamm` using the `contraints()` option:

```
. matrix a=e(b)
. cons def 1 [lns1]i1 = 0
. gllamm y i2-i6, i(id) weight(wt) l(oprob) f(binom) eqs(load) s(het) contr(1) from(a)

number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 40.48688

gllamm model with constraints:

 ( 1)  [lns1]i1 = 0.0

log likelihood = -10114.84022540044

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          i2 |   .0007387   .1116399     0.01   0.995    -.2180716    .2195489
          i3 |  -1.250075    .224931    -5.56   0.000    -1.690931   -.8092181
          i4 |  -1.022133   .2069287    -4.94   0.000    -1.427706   -.6165598
          i5 |  -.8588394   .2049898    -4.19   0.000    -1.260612   -.4570667
          i6 |  -1.988972   .2704984    -7.35   0.000    -2.519139   -1.458805
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.713979   .0442582    38.73   0.000     1.627234    1.800723
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.589511   .0637754    40.60   0.000     2.464514    2.714509
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   3.007872   .0786685    38.23   0.000     2.853685    3.162059
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

    equation for log-standard deviaton:

    i1: 0 (0)
    i2: .02552869 (.06101471)
```

```
    i3: .00171463 (.07983886)
    i4: -.03344029 (.07981058)
    i5: .00183414 (.08008328)
    i6: .56883155 (.07273494)

Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .53880366 (.06639591)

    loadings for random effect 1
    i2: 1.1531179 (.09444011)
    i3: 2.0286322 (.18069937)
    i4: 1.8105319 (.16261248)
    i5: 1.4614725 (.13991799)
    i6: 2.4374615 (.2164596)

------------------------------------------------------------------------------
```

The `(0)` next to the log-standard deviation in the output under "Variance at level 1" reminds us that this parameter was constrained. Item 6 has a much larger estimated scale parameter than the other items and this model fits better than the previous model:

```
. disp chiprob(5,2*(10157.782-10114.84022540044))
4.913e-17
```

Item-specific thresholds

First we fit the model in equation (8.13) without factor loadings, or equivalently, with $\lambda_j = 1$. We allow a different sets of thresholds to be estimated for each item by treating the problem as a mixed response problem. Each response uses the `oprobit` link, but different thresholds will be estimated. We use the `lv()` option to assign each link to a different item:

```
. gllamm y, i(id) weight(wt) l(oprob oprob oprob oprob oprob oprob) lv(item) /*
>   */ f(binom)

number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 9.5386324

gllamm model

log likelihood = -10156.274

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.949536   .0373884    52.14   0.000     1.876256    2.022816
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.943054   .0569112    51.71   0.000      2.83151    3.054598
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   3.363419   .0727558    46.23   0.000      3.22082    3.506017
-------------+----------------------------------------------------------------
```

```
_cut21       |
       _cons |   1.821942   .0356608    51.09   0.000     1.752048    1.891836
-------------+----------------------------------------------------------------
_cut22       |
       _cons |   2.751458   .0515766    53.35   0.000      2.65037    2.852547
-------------+----------------------------------------------------------------
_cut23       |
       _cons |   3.175722   .0644845    49.25   0.000     3.049335    3.302109
-------------+----------------------------------------------------------------
_cut31       |
       _cons |   2.350363   .0441402    53.25   0.000      2.26385    2.436876
-------------+----------------------------------------------------------------
_cut32       |
       _cons |   3.067465   .0604432    50.75   0.000     2.948999    3.185932
-------------+----------------------------------------------------------------
_cut33       |
       _cons |   3.425012   .0735344    46.58   0.000     3.280887    3.569136
-------------+----------------------------------------------------------------
_cut41       |
       _cons |   2.367085   .0444121    53.30   0.000     2.280039    2.454131
-------------+----------------------------------------------------------------
_cut42       |
       _cons |   3.137144   .0627692    49.98   0.000     3.014118    3.260169
-------------+----------------------------------------------------------------
_cut43       |
       _cons |   3.526405    .078913    44.69   0.000     3.371739    3.681072
-------------+----------------------------------------------------------------
_cut51       |
       _cons |   2.487467   .0466838    53.28   0.000     2.395968    2.578966
-------------+----------------------------------------------------------------
_cut52       |
       _cons |   3.326015   .0706762    47.06   0.000     3.187492    3.464538
-------------+----------------------------------------------------------------
_cut53       |
       _cons |   3.775798   .0945832    39.92   0.000     3.590418    3.961177
-------------+----------------------------------------------------------------
_cut61       |
       _cons |   2.088989   .0395998    52.75   0.000     2.011375    2.166603
-------------+----------------------------------------------------------------
_cut62       |
       _cons |   2.579065   .0477213    54.04   0.000     2.485533    2.672597
-------------+----------------------------------------------------------------
_cut63       |
       _cons |   2.826356    .053195    53.13   0.000     2.722095    2.930616
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): 1.0146843 (.05067357)
------------------------------------------------------------------------------
```

The first three parameters represent the thresholds for the first item, the next three those for the second item, etc. The thresholds for the last item are closer together than the other sets of threshold which is consistent with the large scale parameter estimate $\hat{\sigma}_6$ for the last item in the previous model.

Another way of specifying the same model would be using the `thresh()` option. We omit one of the items from the equation for `thresh()` since a constant will automatically be included:

```
. eq thr: i2 i3 i4 i5 i6
```

We will first fit the model without the `thresh()` option to obtain parameter estimates that can be used as starting values. (We would generally recommend this approach with the `thresh()` option.)

```
. qui gllamm y, i(id) weight(wt) l(oprob) f(binom)
. matrix a=e(b)
. gllamm y, i(id) weight(wt) l(oprob) f(binom) thresh(thr) from(a)
number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 20.76315

gllamm model

log likelihood = -10156.274

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
          i2 |  -.1275933   .0378454    -3.37   0.001     -.201769   -.0534176
          i3 |   .4008272   .0435506     9.20   0.000     .3154695    .4861848
          i4 |   .4175487   .0437582     9.54   0.000     .3317843    .5033132
          i5 |   .5379312   .0456097    11.79   0.000     .4485378    .6273245
          i6 |   .1394534   .0402135     3.47   0.001     .0606364    .2182704
       _cons |   1.949534   .0373892    52.14   0.000     1.876253    2.022816
-------------+----------------------------------------------------------------
_cut12       |
          i2 |  -.1915929   .0637849    -3.00   0.003     -.316609   -.0665767
          i3 |   .1244133   .0699024     1.78   0.075    -.0125929    .2614194
          i4 |   .1940915   .0718432     2.70   0.007     .0532814    .3349016
          i5 |   .3829634   .0784386     4.88   0.000     .2292267    .5367002
          i6 |  -.3639866    .060972    -5.97   0.000    -.4834895   -.2444837
       _cons |    2.94305   .0569173    51.71   0.000     2.831494    3.054606
-------------+----------------------------------------------------------------
_cut13       |
          i2 |  -.1876889   .0856696    -2.19   0.028    -.3555982   -.0197796
          i3 |   .0615994   .0917873     0.67   0.502    -.1183004    .2414992
          i4 |   .1629933   .0960767     1.70   0.090    -.0253136    .3513002
          i5 |   .4123862     .10884     3.79   0.000     .1990637    .6257088
          i6 |  -.5370554   .0783063    -6.86   0.000     -.690533   -.3835778
       _cons |    3.36341   .0727657    46.22   0.000     3.220791    3.506028
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): 1.0146821 (.05067342)
------------------------------------------------------------------------------
```

Returning to the previous parameterisation, we now include factor loadings in the model:

```
. gllamm y, i(id) weight(wt) l(oprob oprob oprob oprob oprob oprob) lv(item) /*
>   */ eq(fact) f(binom) trace

number of level 1 units = 38652
```

```
number of level 2 units = 6442

Condition Number = 32.501837

gllamm model

log likelihood = -10113.979
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cut11						
_cons	1.713573	.0442668	38.71	0.000	1.626811	1.800334
_cut12						
_cons	2.606161	.0662968	39.31	0.000	2.476222	2.736101
_cut13						
_cons	2.984135	.0811288	36.78	0.000	2.825125	3.143144
_cut21						
_cons	1.669793	.0457622	36.49	0.000	1.580101	1.759486
_cut22						
_cons	2.530182	.0661797	38.23	0.000	2.400472	2.659892
_cut23						
_cons	2.923071	.0794205	36.80	0.000	2.767409	3.078732
_cut31						
_cons	2.959522	.1504921	19.67	0.000	2.664563	3.254481
_cut32						
_cons	3.826325	.1855404	20.62	0.000	3.462673	4.189978
_cut33						
_cons	4.257651	.205045	20.76	0.000	3.85577	4.659532
_cut41						
_cons	2.830385	.1350866	20.95	0.000	2.56562	3.095149
_cut42						
_cons	3.72574	.1709453	21.79	0.000	3.390693	4.060786
_cut43						
_cons	4.179542	.1925985	21.70	0.000	3.802056	4.557028
_cut51						
_cons	2.568194	.0980636	26.19	0.000	2.375993	2.760395
_cut52						
_cons	3.424813	.1295892	26.43	0.000	3.170823	3.678804
_cut53						
_cons	3.881744	.1531407	25.35	0.000	3.581593	4.181894
_cut61						
_cons	2.096388	.0691691	30.31	0.000	1.960819	2.231957
_cut62						
_cons	2.585095	.0824883	31.34	0.000	2.423421	2.746769
_cut63						
_cons	2.832224	.089877	31.51	0.000	2.656068	3.008379

```
Variances and covariances of random effects
```

```
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .53913465 (.06645437)

    loadings for random effect 1
    i2: 1.1237353 (.09207811)
    i3: 2.0248149 (.1907096)
    i4: 1.8724256 (.17862714)
    i5: 1.4579448 (.13448445)
    i6: 1.3790917 (.11736409)

------------------------------------------------------------------------------
```

The likelihood ratio test indicates that factor loadings are required:

```
. disp chiprob(5,2*(10156.274 -10113.979))
9.175e-17
```

This model is nested in the scaled probit model in (8.13) since the thresholds are freely estimated while the scaled probit model imposes the constraints

$$\kappa_{sj} = (\kappa_{s1} - \beta_j)/\sigma_j, \quad j = 2, \cdots, 6$$

for some β_j and positive σ_j. The unconstrained model has one extra degree of freedom for each set of thresholds and therefore five extra parameters are estimated. However, the likelihood is very close to that of the scaled probit model (-10113.979 compared with -10114.840) so that the former model should be retained.

We will nevertheless develop the current model further to include effects of sex. It is quite possible that the sexes differ in their mean latent delinquency, i.e.

$$u_i = \gamma x_i + \zeta_i$$

where x_i is a dummy variable for girls. Note that there is no intercept in this equation since we are already estimating the three thresholds for each item. The regression of a latent variable on an explanatory variable can be incorporated using the `geqs()` option:

```
matrix a=e(b)
eq f1: sex
. gllamm y, i(id) weight(wt) l(oprob oprob oprob oprob oprob oprob) lv(item)  /*
>   */ eq(fact) f(binom) from(a) geqs(f1)

number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 33.395695

gllamm model

log likelihood = -10075.106

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.560111    .040468    38.55   0.000     1.480796    1.639427
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.439244   .0614223    39.71   0.000     2.318858    2.559629
```

```
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   2.810984   .0761108    36.93   0.000      2.66181    2.960159
-------------+----------------------------------------------------------------
_cut21       |
       _cons |   1.517993   .0428427    35.43   0.000     1.434023    1.601963
-------------+----------------------------------------------------------------
_cut22       |
       _cons |   2.376942   .0622116    38.21   0.000     2.255009    2.498874
-------------+----------------------------------------------------------------
_cut23       |
       _cons |   2.769582   .0753861    36.74   0.000     2.621828    2.917336
-------------+----------------------------------------------------------------
_cut31       |
       _cons |   2.694531   .1349126    19.97   0.000     2.430107    2.958955
-------------+----------------------------------------------------------------
_cut32       |
       _cons |   3.558852   .1693993    21.01   0.000     3.226836    3.890869
-------------+----------------------------------------------------------------
_cut33       |
       _cons |   3.988311   .1889283    21.11   0.000     3.618019    4.358604
-------------+----------------------------------------------------------------
_cut41       |
       _cons |   2.585412   .1177848    21.95   0.000     2.354558    2.816266
-------------+----------------------------------------------------------------
_cut42       |
       _cons |   3.476597   .1515286    22.94   0.000     3.179607    3.773588
-------------+----------------------------------------------------------------
_cut43       |
       _cons |   3.927206   .1726132    22.75   0.000     3.588891    4.265522
-------------+----------------------------------------------------------------
_cut51       |
       _cons |   2.405344   .0923392    26.05   0.000     2.224362    2.586325
-------------+----------------------------------------------------------------
_cut52       |
       _cons |    3.27217   .1244941    26.28   0.000     3.028166    3.516174
-------------+----------------------------------------------------------------
_cut53       |
       _cons |    3.73422    .148731    25.11   0.000     3.442712    4.025727
-------------+----------------------------------------------------------------
_cut61       |
       _cons |   1.941148   .0641503    30.26   0.000     1.815415     2.06688
-------------+----------------------------------------------------------------
_cut62       |
       _cons |    2.43699   .0771453    31.59   0.000     2.285788    2.588192
-------------+----------------------------------------------------------------
_cut63       |
       _cons |   2.687746   .0845165    31.80   0.000     2.522097    2.853395
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .48055439 (.06096913)

    loadings for random effect 1
    i2: 1.1710015 (.09705691)
    i3: 2.1246416 (.20096276)
    i4: 1.9644739 (.18360464)
    i5: 1.565467 (.1469632)
    i6: 1.4882541 (.12793858)


Regressions of latent variables on covariates
------------------------------------------------------------------------------
```

```
    random effect 2 has 1 covariates:
    sex: -.24379616 (.02989614)
------------------------------------------------------------------------------
```

Girls are on average less delinquent, an effect that is significant according to the likelihood ratio test:

```
. disp chiprob(1,2*(10113.979-10075.106))
1.172e-18
```

The model assumes that being a girl affects the responses to the individual items only by affecting overall latent delinquancy u_i, i.e., the effect of being a girl on the jth item is to reduce the latent response by $-0.24\lambda_j$. However, it is quite possible that, even for the same overall delinquency level, girls are more less likely to exhibit certain behaviours than boys. For example, there may be a direct effect of being a girl on the first item in addition to the indirect effect via the latent variable. This effect can be included in the linear predictor as an interaction between sex and the dummy variable for the first item:

```
. matrix a=e(b)
. gen sexi1 = sex*i1
. gllamm y sexi1, i(id) weight(wt) l(oprob oprob oprob oprob oprob oprob) lv(item) /*
>   */ eq(fact) f(binom) from(a) geqs(f1)
number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 27.251127

gllamm model

log likelihood = -10042.639

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
       sexi1 |   .4626439    .060348     7.67   0.000      .344364    .5809239
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.823139   .0579661    31.45   0.000     1.709528    1.936751
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.740513   .0790767    34.66   0.000     2.585525      2.8955
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   3.130098   .0935442    33.46   0.000     2.946755    3.313441
-------------+----------------------------------------------------------------
_cut21       |
       _cons |   1.478449   .0414112    35.70   0.000     1.397285    1.559613
-------------+----------------------------------------------------------------
_cut22       |
       _cons |   2.335236   .0605224    38.58   0.000     2.216615    2.453858
-------------+----------------------------------------------------------------
_cut23       |
       _cons |   2.727335   .0736946    37.01   0.000     2.582896    2.871774
-------------+----------------------------------------------------------------
_cut31       |
       _cons |   2.599276   .1260789    20.62   0.000     2.352166    2.846387
-------------+----------------------------------------------------------------
_cut32       |
       _cons |   3.454004   .1598088    21.61   0.000     3.140785    3.767224
-------------+----------------------------------------------------------------
_cut33       |
```

```
       _cons |   3.878467    .179002    21.67   0.000      3.52763    4.229305
-------------+------------------------------------------------------------------
_cut41       |
       _cons |   2.500363   .1108483    22.56   0.000     2.283104    2.717621
-------------+------------------------------------------------------------------
_cut42       |
       _cons |   3.381748   .1437774    23.52   0.000      3.09995    3.663546
-------------+------------------------------------------------------------------
_cut43       |
       _cons |   3.826931   .1645363    23.26   0.000     3.504445    4.149416
-------------+------------------------------------------------------------------
_cut51       |
       _cons |   2.353478   .0890078    26.44   0.000     2.179026     2.52793
-------------+------------------------------------------------------------------
_cut52       |
       _cons |    3.21905   .1211128    26.58   0.000     2.981674    3.456427
-------------+------------------------------------------------------------------
_cut53       |
       _cons |   3.681461   .1455908    25.29   0.000     3.396109    3.966814
-------------+------------------------------------------------------------------
_cut61       |
       _cons |   1.902898   .0626222    30.39   0.000      1.78016    2.025635
-------------+------------------------------------------------------------------
_cut62       |
       _cons |   2.400715   .0755705    31.77   0.000     2.252599     2.54883
-------------+------------------------------------------------------------------
_cut63       |
       _cons |    2.65268   .0829613    31.97   0.000     2.490079    2.815281
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .63344967 (.07759424)

    loadings for random effect 1
    i2: 1.0029452 (.08274042)
    i3: 1.8036891 (.16951336)
    i4: 1.6696892 (.15472651)
    i5: 1.3477244 (.12501426)
    i6: 1.295613 (.10910991)


Regressions of latent variables on covariates
------------------------------------------------------------------------------

    random effect 2 has 1 covariates:
    sex: -.34808251 (.03932434)
------------------------------------------------------------------------------
```

For the same delinquency level, girls tend to have higher scores on the first item. This effect is again significant:

```
. disp chiprob(1,2*(10075.106-10042.639))
7.745e-16
```

The difference in mean latent response for item 1 between an average girl and an average boy is

$$-0.348 \times 1 + 0.463$$

Finally, it could be that the effect of being a girl is not the same for each threshold. For example, after taking into account overall delinquency, girls may be more likely than boys to be in category 2

but not to be in category 3. This can be modelled using the `thresh()` option. One equation must be specified for each response, so if we do not want items 2 to 6 to have thresholds that depend on sex, we must set up equations with no variables on the right hand side:

```
. eq sex: sex
. eq n:
. gllamm y, i(id) weight(wt) l(oprob oprob oprob oprob oprob oprob) lv(item)  /*
>   */ eq(fact) f(binom) from(a) geqs(f1) thr(sex n n n n n)

number of level 1 units = 38652
number of level 2 units = 6442

Condition Number = 27.425758

gllamm model

log likelihood = -10041.861

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
         sex |  -.4532846   .0614316    -7.38   0.000    -.5736884   -.3328809
       _cons |   1.818512   .0581802    31.26   0.000     1.704481    1.932543
-------------+----------------------------------------------------------------
_cut12       |
         sex |  -.4877739   .0963228    -5.06   0.000    -.6765631   -.2989846
       _cons |   2.755284   .0903551    30.49   0.000     2.578192    2.932377
-------------+----------------------------------------------------------------
_cut13       |
         sex |  -.5984464    .129981    -4.60   0.000    -.8532046   -.3436882
       _cons |   3.211047   .1189575    26.99   0.000     2.977895    3.444199
-------------+----------------------------------------------------------------
_cut21       |
       _cons |    1.47873    .041411    35.71   0.000     1.397566    1.559894
-------------+----------------------------------------------------------------
_cut22       |
       _cons |   2.335549   .0605201    38.59   0.000     2.216932    2.454166
-------------+----------------------------------------------------------------
_cut23       |
       _cons |   2.727699   .0736939    37.01   0.000     2.583261    2.872136
-------------+----------------------------------------------------------------
_cut31       |
       _cons |   2.598953   .1259928    20.63   0.000     2.352011    2.845894
-------------+----------------------------------------------------------------
_cut32       |
       _cons |   3.453649   .1597085    21.62   0.000     3.140626    3.766672
-------------+----------------------------------------------------------------
_cut33       |
       _cons |   3.878291   .1789169    21.68   0.000      3.52762    4.228961
-------------+----------------------------------------------------------------
_cut41       |
       _cons |   2.499882   .1107762    22.57   0.000     2.282764    2.716999
-------------+----------------------------------------------------------------
_cut42       |
       _cons |   3.381226   .1437036    23.53   0.000     3.099572     3.66288
-------------+----------------------------------------------------------------
_cut43       |
       _cons |   3.826555   .1644752    23.27   0.000     3.504189     4.14892
-------------+----------------------------------------------------------------
_cut51       |
       _cons |   2.353664   .0889789    26.45   0.000     2.179268    2.528059
-------------+----------------------------------------------------------------
_cut52       |
       _cons |   3.219422    .121096    26.59   0.000     2.982078    3.456766
-------------+----------------------------------------------------------------
_cut53       |
```

```
       _cons |   3.682137   .1456064    25.29   0.000     3.396754     3.96752
-------------+----------------------------------------------------------------
_cut61       |
       _cons |   1.903018   .0626056    30.40   0.000     1.780313    2.025723
-------------+----------------------------------------------------------------
_cut62       |
       _cons |   2.400811   .0755498    31.78   0.000     2.252736    2.548886
-------------+----------------------------------------------------------------
_cut63       |
       _cons |   2.652798    .082941    31.98   0.000     2.490237     2.81536
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (id)

    var(1): .63355899 (.07755586)

    loadings for random effect 1
    i2: 1.0026925 (.08269602)
    i3: 1.8025142 (.16934718)
    i4: 1.6683532 (.15462029)
    i5: 1.3472172 (.12492251)
    i6: 1.2950411 (.10903563)


Regressions of latent variables on covariates
------------------------------------------------------------------------------

    random effect 2 has 1 covariates:
    sex: -.34740644 (.03929033)
------------------------------------------------------------------------------
```

There is little evidence for a differential effect of sex on the three thresholds with a small change in log-likelihood and coefficients of sex that are similar across thresholds.

Chapter 9

Nominal or polytomous responses and rankings

9.1 Multinomial logit model for nominal data

Let a index the A possible categories of the response variable; it is convenient to think of these categories as alternatives and the response as a choice among alternatives even if the response does not strictly represent a choice. We will define multinomial logit models by specifying the 'linear predictor' V^a, $a = 1, \cdots, A$ so that the multinomial probability of response category f (the probability that f is chosen) is

$$\Pr(f) = \frac{\exp(V^f)}{\sum_{a=1}^{A} \exp(V^a)} \tag{9.1}$$

This probability model can also be derived by assuming that associated with each alternative is an unobserved 'utility' U^a (latent response) and that the alternative with the highest utility is selected. Depending on the situation, utility could means attractivess, usefulness (voting/purchasing), or cost-effectiveness (clinical treatments) of the alternative. The utility is modelled as

$$U^a = V^a + \epsilon^a. \tag{9.2}$$

Alternative f is selected if

$$U^f > U^g \text{ for all } g \neq f \tag{9.3}$$

or, equivalently,

$$U^f - U^g = V^f - V^g + (\epsilon^f - \epsilon^g) > 0. \tag{9.4}$$

If the error term ϵ^a has an extreme value distribution of type I (Gumbel), then the differences $(\epsilon^f - \epsilon^g)$ have a logistic distribution and equation (9.1) follows (McFadden, 1973).

For subject-specific covariates, a different coefficient vector $\mathbf{g}^a$ is estimated for each alternative except a reference alternative:

$$V^a = \mathbf{g}^{a\prime}\mathbf{x} \tag{9.5}$$

Although the multinomial logit or *polytomous logistic regression* model usually only includes subject specific covariates, we can also include alternative specific covariates in `gllamm`. This is done by expanding the data so that for each subject there is one record for each alternative available to that subject (different alternative sets for different individuals are possible) and creating an indicator for the variable actually selected. Alternative (and subject) specific variables or random effects can then easily be specified. When running `gllamm`, the `expanded()` option must be used to specify the clusters of observations (subjects) representing a single alternative set so that `gllamm` computes

a single likelihood contribution for each alternative set equal to the multinomial probability in equation (9.1).

Consider the example where the outcome is the choice of mode of transport used for commuting. Person 1 can choose between a train, bus or car, and person 2 can only choose between a bus or a car because there is no train available to him. The alternative sets therefore have different sizes. We know the people's ages and the cost of their journey from home to work for each mode of transport (an alternative and subject specific covariate). The data would have to be set up as follows:

person	age	mode	cost	choice
1	23	train	2	1
1	23	bus	1.6	0
1	23	car	2	0
2	30	bus	0.8	0
2	30	car	1.2	1

where 'choice' indicates the mode of transport chosen.

9.2 Multinomial logit model for rankings

Rankings are orderings of alternatives (parties, clinical treatments, brands) according to preference or some other characteristic. (A nominal response represents an incomplete ranking where only the first choice is specified.) As in the previous section, we assume that associated with each alternative a there is a utility U^a

$$U^a = V^a + \epsilon^a \tag{9.6}$$

and again assume that ϵ^a has an extreme value distribution of type I (Gumbel). Let r^s be the alternative with rank s. Then the ranking $R = (r^1, r^2, \cdots r^A)$ is obtained if

$$U^{r^1} > U^{r^2} > \cdots > U^{r^A} \tag{9.7}$$

and the probability of a ranking R is (Luce, 1959)

$$\Pr(R) = \frac{\exp(V^{r^1})}{\sum_{s=1}^{A}\exp(V^{r^s})} \times \frac{\exp(V^{r^2})}{\sum_{s=2}^{A}\exp(V^{r^s})} \times \cdots \times \frac{\exp(V^{r^A})}{\sum_{s=A-1}^{A}\exp(V^{r^s})}. \tag{9.8}$$

This probability can be interpreted as arising from a sequential choice process where the subject initially makes a first choice among all alternatives. In the second 'stage', a first choice is made among all alternatives except the first since this is no longer available. At each subsequent stage, a first choice is made among the alternatives still remaining at that stage. The likelihood looks like the partial likelihood of Cox's regression model where the ranks are the survival times and the alternative sets remaining at each stage represent the 'risk sets'. We can expand the data to alternative sets and then estimate the model in the same way as for the first choice case. The likelihood contribution of each alternative set is the same as that of a subject in the first choice case (with different alternative sets for different 'individuals'). If these alternative sets are specified in `gllamm` using the `expanded()` option, their product is automatically evaluated yielding the expression in equation (9.8).

Consider the commuting example given in the previous section, but this time the respondents could rate the modes of transport in order of preferance. Person 1 ranks the modes in the order train, bus, car and person 2 ranks the two modes available to him in the order bus, car. The data would have to be expanded or "exploded" as follows:

person	age	stage	alternative set	mode	cost	choice
1	23	1	1	train	2	1
1	23	1	1	bus	1.6	0
1	23	1	1	car	2	0
1	23	2	2	bus	1.6	1
1	23	2	2	car	2	0
2	30	1	3	bus	0.8	0
2	30	1	3	car	1.2	1

Here, person 1 has two 'decision stages' - in the first, all three alternatives are available and in the second, a choice is made between bus and car. Person 2 only has one 'decision stage' because only one alternative is left after making the first choice. It is clear how incomplete rankings can be handled, where individuals only rank the top few alternatives. Multilevel models for first choice and ranking data are discussed in Skrondal and Rabe-Hesketh (2001).

9.3 Nominal response: Multinomial logit with a random intercept

In this section we use the Junior School Project data used to illustrate the multilevel multinomial logit model in the MLwiN Advanced Macros manual (Yang *et al.*, 1999).

The teachers' rating of pupils' behaviour is available on 3939 pupils (j) in 48 schools (i). Although the rating is ordinal with scores 1,2,3 representing the top 25%, the middle 50%, and the bottom 25%, respectively, we will repeat the analysis of the Advanced Macros manual so that we can compare the estimates using quadrature with the MQL/PQL estimates used in MLwiN.

The linear predictor, or utility, includes a subject-specific covariate, the sex of subject j in school i, x_{ij} as well as random intercepts for school, γ^a_{i0}:

$$V^a_{ij} = g^a_0 + g^a_1 x_{ij} + \gamma^a_{0i} \tag{9.9}$$

where all effects are set to 0 for $a = 1$, i.e. $g^1_0 = g^1_1 = 0$ and $\gamma^1_{0i} = 0$, making the first alternative the 'reference category'. There are therefore two random effects, γ^2_{0i} and γ^3_{0i} for alternative 2 and 3 and these random effects will be assumed to be correlated.

9.3.1 Data preparation

The data are available as an ASCII file *jspmix.dat*. We read the file using `infile`:

```
infile scy3 id sex stag ravi fry3 tby using jspmix.dat, clear
```

Here `scy3` is the school index, `tby` is the response variable and `sex` will be used as an explanatory variable. (The other variables will not be used). Since many pupils in the same school are likely to have the same response and sex, we can collapse the data and form level 1 weights to speed up the estimation.

```
gen cons=1
collapse (count) wt1=cons, by(scy3 sex tby )
```

The level 1 weight variable is `wt1` and we will have to specify `wt` in the `weight()` option when running `gllamm`. The first 12 observations now are (use `sort scy3 sex tby` if the observations are in a different order):

```
          scy3        sex        tby        wt1
 1.          1          0          1          8
 2.          1          0          2          2
 3.          1          0          3          2
 4.          1          1          1          3
 5.          1          1          2          8
 6.          1          1          3          4
 7.          2          0          1          3
 8.          2          0          2          2
 9.          2          0          3          4
10.          2          1          2          4
11.          2          1          3          4
12.          3          0          1          3
```

For example, 8 individuals in the first school had `sex` = 0 and `tby` = 1.

We can run the model without random effects using Stata's `mlogit` command with frequency weights and using the response `tby`=1 as the baseline category:

```
. mlogit tby sex [fweight=wt1], base(1)

Iteration 0:   log likelihood = -1349.1298
Iteration 1:   log likelihood = -1332.0463
Iteration 2:   log likelihood = -1331.9206
Iteration 3:   log likelihood = -1331.9206

Multinomial regression                          Number of obs   =       1313
                                                LR chi2(2)      =      34.42
                                                Prob > chi2     =     0.0000
Log likelihood = -1331.9206                     Pseudo R2       =     0.0128

------------------------------------------------------------------------------
         tby |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
2            |
         sex |   .4227768   .1368457     3.09   0.002     .1545643    .6909894
       _cons |   .5381711   .0885475     6.08   0.000     .3646211    .7117211
-------------+----------------------------------------------------------------
3            |
         sex |   .9436537   .1632866     5.78   0.000     .6236178     1.26369
       _cons |  -.5460938   .1161788    -4.70   0.000    -.7737999   -.3183876
------------------------------------------------------------------------------
(Outcome tby==1 is the comparison group)
```

The `gllamm` command can be used to obtain the same estimates using the mlogit link and the binomial family:

```
gllamm tby sex, i(scy3) init base(1) link(mlogit) family(binom) weight(wt) trace
```

Here we used the `init` option to obtain the 'initial estimates' where all random components are set to 0. (Here the option `i(scy3)` serves no purpose but is used because the `i()` 'option' is required.)

Since different random intercepts apply to alternatives 2 and 3, we will expand the data so that there is one record for each alternative for each observation in the current dataset. A dummy variable, `chosen` will indicate which of the alternatives was selected. Note that this type of expansion also allows alternative specific covariates to be included in the linear predictor as well as allowing different alternative sets (sets of possible response categories) for different individuals.

We create an identifier, `patt`, for the records in the current dataset which represent unique combinations (or patterns) of `scy3`, `sex` and `tby`.

```
sort school sex tby
gen patt=_n
```

We now need to expand the data. This is easy because all individuals choose from the same full set of three alternatives. First we replace each record with three replicates of itself and define a new variable `alt` which takes on the possible values of `tby` for each value of `patt`. The variable `chosen` indicates which of the values in `alt` equals the response actually given.

```
expand 3
sort patt
qui by patt: gen alt=_n
gen chosen=alt==tby
```

The data now look like this:

```
. sort patt alt
. list scy3 patt sex  alt chosen tby in 1/21
```

	scy3	patt	sex	alt	chosen	tby
1.	1	1	0	1	1	1
2.	1	1	0	2	0	1
3.	1	1	0	3	0	1
4.	1	2	0	1	0	2
5.	1	2	0	2	1	2
6.	1	2	0	3	0	2
7.	1	3	0	1	0	3
8.	1	3	0	2	0	3
9.	1	3	0	3	1	3
10.	1	4	1	1	1	1
11.	1	4	1	2	0	1
12.	1	4	1	3	0	1
13.	1	5	1	1	0	2
14.	1	5	1	2	1	2
15.	1	5	1	3	0	2
16.	1	6	1	1	0	3
17.	1	6	1	2	0	3
18.	1	6	1	3	1	3
19.	2	7	0	1	1	1
20.	2	7	0	2	0	1
21.	2	7	0	3	0	1

9.3.2 Parameter estimation

We now use `alt` as the dependent variable and must use the `expanded()` option to indicate that the data are in expanded form. The arguments for this option are the identifier of the records in the original, unexpanded dataset, here `patt`, the indicator variable for the alternative that was selected and either `o` or `m`. The `o` option (stands for "one") indicates that one fixed coefficient is to be estimated for each explanatory variable and requires dummy variables to be used to estimate separate parameters for alternatives 2 and 3. The `m` option (stands for "many") indicates that $A-1$ parameters are to be estimated for each explanatory variable.

In either case, we must specify equations for the random intercepts γ_0^2 and γ_0^3 using appropriate dummy variables:

```
tab alt, gen(a)
eq a2: a2
eq a3: a3
```

The shorter way of running `gllamm` is to use `m` in the `expand()` option:

```
gllamm alt sex, expand(patt chosen m) i(scy3) lin(mlogit) family(binom)/*
*/ nrf(2) eqs(a2 a3) nip(4) weight(wt) trace
```

but we will define the appropriate dummy variables and use `o` in the `expand()` option:

```
gen a2sex = a2*sex
gen a3sex = a3*sex
gllamm alt a2 a3 a2sex a3sex, nocons expand(patt chosen o) i(scy3) /*
         */ lin(mlogit) family(binom) nrf(2) eqs(a2 a3) nip(4) weight(wt) /*
         */ trace
```

Here the `nocons` option is used since we do not want to include an overall constant in V^a (the constant would cancel out in equation (9.1) and is therefore not identified.) After estimating the model with 4 quadrature points, (log-likelihood =-1300.9504) we use the commands

```
matrix a=e(b)
gllamm alt a2 a3 a2sex a3sex, nocons expand(patt chosen o) i(scy3) /*
         */ lin(mlogit) family(binom) nrf(2) eqs(a2 a3) nip(8) weight(wt) /*
         */ from(a) trace
```

to estimate the model with 8 quadrature points (log-likelihood = -1299.4815) and finally, we estimate the model with 12 quadrature points:

```
. gllamm alt a2 a3 a2sex a3sex, nocons expand(patt chosen o) i(scy3) /*
>         */ lin(mlogit) family(binom) nrf(2) eqs(a2 a3) from(a) nip(12) /*
>         */ weight(wt) trace

General model information
------------------------------------------------------------------------------
dependent variable:         alt
family:                     binom
link:                       mlogit
denominator:                1
equation for fixed effects  a2 a3 a2sex a3sex

Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (scy3) equation(s):
   (2 random effect(s))


   diagonal element of cholesky decomp. of covariance matrix
   scy31 : a2

   diagonal element of cholesky decomp. of covariance matrix
   scy32 : a3
```

```
   off-diagonal elements
   scy32_1: _cons

number of level 1 units = 3939
number of level 2 units = 48

>>>>> iterations omitted

Condition Number = 4.9502936

gllamm model

log likelihood = -1299.6698

------------------------------------------------------------------------------
         alt |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          a2 |   .5925784   .1414131     4.19   0.000     .3154139    .8697428
          a3 |  -.5694212   .1805909    -3.15   0.002    -.9233729   -.2154695
       a2sex |   .5464019   .1456529     3.75   0.000     .2609274    .8318764
       a3sex |   1.101336   .1747446     6.30   0.000     .7588433    1.443829
------------------------------------------------------------------------------


Variances and covariances of random effects
------------------------------------------------------------------------------

***level 2 (scy3)

    var(1): .49436301 (.18268588)
    cov(1,2): .54890185 (.19080157) cor(1,2): .90634176

    var(2): .74192403 (.24617685)
------------------------------------------------------------------------------
```

There are several reasons for gradually increasing the number of quadrature points: (1) estimation with 12 quadrature points per dimension is very slow and it is good to have some preliminary results quickly to make sure the model was specified correctly (2) the 12 quadrature point estimation will require fewer iterations if the starting values are good (from a previous run with fewer quadrature points) and (3) we need values of the maximised log-likelihood and parameter estimates for different numbers of quadrature points to assess the adequacy of the approximations (see `quadchk` in Stata Reference manual).

The effect of `sex` is to increase the odds of alternatives 2 and 3 compared with alternative 1. To obtain the odds ratios, use the `eform` option when estimating the parameters or issue the command

```
gllamm, eform
```

after parameter estimation.

There is strong evidence for between school variation with a change in log-likelihood of 32 when the two random effects were introduced (3 parameters). The two random effects are highly correlated.

9.3.3 Comparison with MLwiN

Table 9.1 shows the parameter estimates obtained in MLwiN, using second order MQL (PQL did not converge) next to those obtained in `gllamm`. There are large differences in the school level variance estimates between `gllamm` and MLwiN.

For binary responses, MQL and PQL are known to underestimate the variances of the random effects. The bias is greater for MQL than for PQL. We therefore carried out the following simulation to find out whether `gllamm` overestimated the variances or MLwin underestimated the variances. We replicated each school 10 times, each with the same number of pupils and number of boys as in the data. We then simulated the behaviour rating using the MLwiN estimates as the 'true parameters' and estimated models using both `gllamm` and MLwiN. Using the `gllamm` estimates as 'true parameters', we then resimulated the responses and again estimated the parameters using both `gllamm` and MLwiN. The results are shown in Table 9.2. For both simulations, the `gllamm` parameter estimates are closer to the true values than the MQL values (PQL did not converge).

9.4 A latent class model for rankings

Croon (1989) describe a latent class analysis of rankings. In 1974/1975, over 2000 German respondents rated four political goals according to their desirability:

1. Maintain order in the nation
2. Give people more say in decisions of the government
3. Fight rising prices
4. Protect freedom of speech

The purpose of this ranking task was to investigate value orientations which may be classifiable as materialistic or post-materialistic; materialists would be expected to give preference to goals 1 and 3 whereas post materialists would be expected to prefer goals 2 and 4. The heterogeneity in value orientations can be modelled by assuming that subjects' 'utilities' for the political goals vary randomly from the overall mean, i.e.,

The model is a latent class model with

$$V_i^a = g^a + \gamma_i^a \quad a = 1, 2, 3 \tag{9.10}$$

Table 9.1: MLwin and gllamm estimates for Junior School Project data

		MLwiN estimates (2nd order MQL)		gllamm estimates (12pts)	
		estimate	SE	estimate	SE
cat. 2	cons	0.468	0.094	0.593	0.141
	boy	0.445	0.112	0.546	0.181
cat. 3	cons	-0.631	0.123	-0.569	0.145
	boy	0.973	0.137	1.101	0.175
	var(2)	0.115	0.054	0.494	0.183
	var(3)	0.189	0.082	0.741	0.246
	cov(2,3)	0.019	0.048	0.549	0.191

Table 9.2: Simulations of multinomial responses with estimated parameters using `gllamm` and MLwiN.

			gllamm estimates (12pts)		MLwiN estimates (2nd order MQL)	
MLwiN param.		True value	estimate	SE	estimate	SE
cat. 2	cons	0.468	0.456	0.033	0.468	0.029
	boy	0.445	0.408	0.043	0.461	0.035
cat. 3	cons	-0.631	-0.597	0.043	-0.588	0.038
	boy	0.973	0.948	0.052	0.971	0.043
	var(2)	0.115	0.117	0.022	0.105	0.016
	var(3)	0.189	0.169	0.032	0.161	0.024
	cov(2,3)	0.019	0.021	0.020	-0.114	0.017
gllamm param.						
cat. 2	cons	0.593	0.587	0.056	0.481	0.030
	boy	0.546	0.496	0.046	0.505	0.035
cat. 3	cons	-0.569	-0.560	0.057	-0.656	0.038
	boy	1.101	1.102	0.055	1.119	0.043
	var(2)	0.494	0.519	0.053	0.119	0.017
	var(3)	0.741	0.740	0.075	0.153	0.023
	cov(2,3)	0.549	0.573	0.058	0.025	0.015

and

$$V^4 = 0. \tag{9.11}$$

If subjects fall into latent groups or types (e.g. materialistic and post-materialistic), then this can be modelled by assuming that the random effects $(\gamma_i^1,\gamma_i^2,\gamma_i^3)$ take on discrete values (z_{1r},z_{2r},z_{3r}), $r = 1,\cdots R$ with probabilities π_r.

If `gllamm` is used with the `ip(f)` option, the g^a represent the mean locations and the γ_i^a represent deviations from the mean. If we use the `ip(fn)` option instead, the γ_i^a are not centered around their means and the constants g^a become redundant.

9.4.1 Data preparation

The rankings in the data are represented by four variables, `item1` to `item4`, where `item1` specifies the most preferred alternative (goal in this case), `item2` specifies the second preference, etc. The data contain each of the 24 ($4 \times 3 \times 2$) rankings and the number of times they occurred.

We read the data and stack the alternatives into a single variable, `item`, defining the variable `rank` which contains the ranking of the alternatives:

```
infile item1 item2 item3 item4 wt2 using materia.dat, clear
gen patt=_n
reshape long item, i(patt) j(rank)
```

The data now look like this:

```
. sort patt rank
. list patt rank item wt2 in 1/8

        patt       rank       item        wt2
 1.        1          1          1        137
 2.        1          2          2        137
 3.        1          3          3        137
 4.        1          4          4        137
 5.        2          1          1         29
 6.        2          2          2         29
 7.        2          3          4         29
 8.        2          4          3         29
```

137 individuals gave the rank order 1,2,3,4 and 29 individuals the order 1,2,4,3. We now need to expand the data further. Regarding the ranks as sequential decisions, where in each stage the best remaining alternative is selected, we need to expand the data to 'alternative sets'. Analogously to risk sets in survival analysis representing all those who are still available (and can still fail) at a given time, the alternative sets represent all alternatives that are still available at a given stage (and can still be chosen). Using this analogy, we can simply use Stata's `stsplit` command (available in Stata 7) for expanding survival data to risk set data (see Chapter 7). Here the 'survival times' are the rankings in the variable `rank` and we need to stratify by `patt`. The 'failure' indicator, which we will call `chosen`, is always 1 because a choice was made at each stage. The individual records in the current dataset correspond to 'individuals' in the survival setting. We first need to define the data as surival data using `stset`. We can then expand the data using `stsplit`:

```
gen id=_n
gen chosen = 1
stset rank, fail(chosen) id(id)
stsplit, at(failures) strata(patt) riskset(set)
```

The data now look like this:

```
. sort patt set item
. list patt set rank item chosen wt2 in 1/20

        patt        set       rank       item     chosen        wt2
 1.        1          1          1          1          1        137
 2.        1          1          1          2          0        137
 3.        1          1          1          3          0        137
 4.        1          1          1          4          0        137
 5.        1          2          2          2          1        137
 6.        1          2          2          3          0        137
 7.        1          2          2          4          0        137
 8.        1          3          3          3          1        137
 9.        1          3          3          4          0        137
10.        1          4          4          4          1        137
11.        2          5          1          1          1         29
12.        2          5          1          2          0         29
13.        2          5          1          3          0         29
14.        2          5          1          4          0         29
15.        2          6          2          2          1         29
16.        2          6          2          3          0         29
17.        2          6          2          4          0         29
18.        2          7          3          3          0         29
19.        2          7          3          4          1         29
20.        2          8          4          3          1         29
```

At `set` (alternative set) 1 for `patt` 1, all four items were available and the fist was chosen as indicated by the variable `chosen`. At `set` 2, the first item was no longer available, so only items 2, 3 and 4 are represented and the second is chosen. At `set` 4, only one alternative is left and this observation is redundant (the multinomial probability for that 'alternative set' would be 1). We therefore drop all observations with `rank` equal to 4:

```
. drop if rank==4
(24 observations deleted)
```

We need to define dummy variables for the alternatives so that we can specify the model in terms of alternative specific parameters:

```
tab item, gen(alt)
```

9.4.2 Parameter estimation

In equation (9.10), we need a constant g^a and a random effect γ^a for each alternative except the last. The constants can be included by listing the dummy variables for the first three alternatives as explanatory variables and using the `nocons` option. The random effects are included by defining three equations, one for each alternative, specifying that there are three random effects at level 2 using the `nrf()` option and listing the three equations in the `eqs()` option. The `ip(f)` option is used to specify discrete random effects and initially, we will estimate the model with two latent classes, i.e. using the `nip(2)` option:

```
. eq alt1: alt1
. eq alt2: alt2
. eq alt3: alt3

. gllamm item alt1 alt2 alt3, expand(set chosen o) i(patt) link(mlogit) /*
>    */ family(binom) nrf(3) eqs(alt1 alt2 alt3) nocons weight(wt) nip(2) /*
>    */ ip(f) trace

General model information
------------------------------------------------------------------------------
dependent variable:         item
family:                     binom
link:                       mlogit
denominator:                1
equation for fixed effects  alt1 alt2 alt3

Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (patt) equation(s):
   (3 random effect(s))


class 1

   location for random effect 1
   z2_1_1: alt1

   location for random effect 2
```

```
   z2_2_1: alt2

   location for random effect 3
   z2_3_1: alt3

   log odds for level 2
   p2_1: _cons

>>> output omitted

Initial values for fixed effects

>>> output omitted

                                                  log likelihood = -6427.0497
------------------------------------------------------------------------------
------------------------------------------------------------------------------
        item |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        alt1 |   1.164864   .0411781    28.29   0.000     1.084156    1.245572
        alt2 |   .2106669   .0387733     5.43   0.000     .1346727    .2866611
        alt3 |     1.2767   .0412705    30.93   0.000     1.195811    1.357589
------------------------------------------------------------------------------

>>> output omitted

number of level 1 units = 20358
number of level 2 units = 2262

Condition Number = 10.249569

gllamm model

log likelihood = -6311.6859

------------------------------------------------------------------------------
        item |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        alt1 |   1.362889   .0584633    23.31   0.000     1.248303    1.477475
        alt2 |   .2561528   .0408197     6.28   0.000     .1761477    .3361579
        alt3 |   1.439707   .0548395    26.25   0.000     1.332224    1.547191
------------------------------------------------------------------------------


Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (patt)
    prob: 0.2061, 0.7939

    loc1: -2.2346, .57995
  var(1): 1.2959723
cov(1,2): -.10509489
cov(1,3): .95773595

    loc2: .18121, -.04703
  var(2): .00852251
cov(2,3): -.07766613
```

```
    loc3: -1.6514, .42859
  var(3): .70777605
------------------------------------------------------------------------------
```

On average, the materialistic goals were preferred (the coefficients of `alt1` and `alt3` are larger than that of `alt2` and 0 (the implied coefficient of `alt4`)). About 21% of the population fall into the post-materialistic category with negative effects for goals 1 and 3 (latent class 1) and about 79% of the population fall into the materialistic category (latent class 2).

Croon (1989) assesses the adequacy of different numbers of latent classes using the deviance. The deviance is twice the difference in log-likelihoods between a given model and the full or saturated model. The latter can be obtained from the original data by estimating the probability of each of the 24 possible rankings:

```
. infile item1 item2 item3 item4 wt2 using materia.dat, clear
(24 observations read)

. summ wt2

    Variable |     Obs        Mean   Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         wt2 |      24       94.25   97.60312         21        330

. disp r(sum)
2262

. gen l=wt2*ln(wt2/2262)

. summ l

    Variable |     Obs        Mean   Std. Dev.       Min        Max
-------------+--------------------------------------------------------
           l |      24   -261.2302   171.8756  -635.2209  -98.26913

. disp r(sum)
-6269.5248
```

In the initial `gllamm` output for the two class solution, the 'fixed effects estimates' are given. These are the estimates when the random effects are set to 0 and therefore correspond to the one class solution. Therefore, the deviances for the one class and two class solutions are

```
. disp 2*(6427.0497 - 6269.5248)
315.0498

. disp 2*(6311.6859 - 6269.5248)
84.3222
```

which agrees with the values given in Croon.

The locations in the output represent the deviations of the individual latent classes from the means. To stop `gllamm` from centering the locations around their means, we can use the `ip(fn)` option (and specify no fixed effects):

```
. gllamm item, expand(set chosen o) i(patt) link(mlogit) family(binom)/*
>   */ nrf(3) eqs(alt1 alt2 alt3) nocons weight(wt) nip(2) ip(fn) trace
```

```
General model information
------------------------------------------------------------------------------
dependent variable:         item
family:                     binom
link:                       mlogit
denominator:                1
equation for fixed effects  none

Random effects information for 2 level model
------------------------------------------------------------------------------


***level 2 (patt) equation(s):
   (3 random effect(s))


class 1

   location for random effect 1
   z2_1_1: alt1

   location for random effect 2
   z2_2_1: alt2

   location for random effect 3
   z2_3_1: alt3

   log odds for level 2
   p2_1: _cons

class 2

   location for random effect 1
   z2_1_2: alt1

   location for random effect 2
   z2_2_2: alt2

   location for random effect 3
   z2_3_2: alt3

>>> output omitted

number of level 1 units = 20358
number of level 2 units = 2262

Condition Number = 9.3329566

gllamm model

log likelihood = -6311.6859

No fixed effects


Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (patt)
```

```
    prob: 0.2061, 0.7939

    loc1: -.87172, 1.9428
  var(1): 1.295966
cov(1,2): -.10509344
cov(1,3): .95773294

    loc2: .43736, .20912
  var(2): .00852232
cov(2,3): -.07766519

    loc3: -.2117, 1.8683
  var(3): .70777505
------------------------------------------------------------------------------
```

The log-likelihood is as before but the locations are not centred anymore making them easier to interpret.

We now obtain the three latent class solution using the Gateaux derivative method. Here, the log-likelihood is evaluated at the parameter estimates of the two point solution while a third point with a very low probability is added and moved through a fine 3-dimensional grid of locations (searching the range -5 to 5 in 20 steps in each dimension):

```
. matrix a=e(b)
. local k=e(k)
. local ll=e(ll)
. noi cap noi gllamm item, expand(set chosen o) i(patt) link(mlogit)   /*
>  */ family(binom) nrf(3) eqs(alt1 alt2 alt3) nocons weight(wt) from(a) /*
>  */ nip(3) ip(fn) trace gateaux(-5 5 20) lf0(`k' `ll')

>>> output omitted

number of level 1 units = 20358
number of level 2 units = 2262

Condition Number = 11.369391

gllamm model

log likelihood = -6281.3611

No fixed effects


Probabilities and locations of random effects
------------------------------------------------------------------------------

***level 2 (patt)
    prob: 0.225, 0.3211, 0.4538

    loc1: -.76144, 3.1448, 1.8385
  var(1): 2.0418146
cov(1,2): -.19209441
cov(1,3): .83004357

    loc2: .55538, .21003, .17141
  var(2): .0238936
cov(2,3): -.16114594
```

```
    loc3: -.08917, 1.1827, 2.9561
  var(3): 1.5224173
------------------------------------------------------------------------------
```

The deviance now is

```
. disp 2*(6281.3611 - 6269.5248)
23.6726
```

Croon's three class estimates are given in his Table 3 where he uses yet another parameterisation. Instead of fixing $\gamma^4 = 0$, he uses the constraint $\sum_a \gamma_a = 0$. For the largest class with probability 0.45, this gives the following four locations:

```
. disp 1.8385-(1.8385+.17141+2.9561)/4
.5969975

. disp .17141-(1.8385+.17141+2.9561)/4
-1.0700925

. disp 2.9561-(1.8385+.17141+2.9561)/4
1.7145975

. disp -(1.8385+.17141+2.9561)/4
-1.2415025
```

which (almost) agrees with Croon's result of 0.59,-1.07,1.73,-1.25.

Appendix A

A quick introduction to Stata

This section briefly discusses the most important Stata commands used in this manual, mostly for preparing the data for `gllamm`. See Rabe-Hesketh and Everitt (2000) for a more complete introduction to Stata.

The most important basic commands are `use` or `infile` for reading data and `save` for saving data, `list` for listing data, `generate`, `replace`, `egen` and `recode` for transforming variables and `drop` and `keep` for dropping observations. Basic data summary commands are `tabulate`, `table` and `summarize` and basic estimation commands are `regress`, `logit`, `glm`, etc. If you are not already familiar with these commands, look these up using Stata's help or in the reference manual. Also look in the Stata User's Guide under Estimation and Post-estimation commands.

Many stata commands use variable lists (abbreviated `varlist` in syntax descriptions). These are just lists of variables separated by spaces. For example, to regress `y` on `x1`, `x2`, `x3`, `sp`, `st` and `houses`, use

```
regress y x1 x2 x3 sp st houses
```

Variables can be abbreviated as long as this is unambiguous (i.e. only one variable in the data has the same abbreviation). Variable lists can also be abbreviated. If there are no other variables in the dataset, the following commands are all equivalent:

```
regress y x1 x2 x3 sp st houses
regress y x1 x2 x3 sp st hous
regress y x1-x3 sp st hous
regress y x1-x3 s* hous
```

Numeric expressions (e.g. used in `generate`) look like they do in most packages, e.g.,

```
gen x = y + z
replace x = (y - z)*5
replace x = 10/x
replace x = x^2
replace x = sqrt(x/2)
```

where `+ - * /` and `^` are the plus, minus, times, divide and power operators, respectively, and `sqrt()` is the square root. See help for `functions` to find out about more functions.

Almost all Stata commands can be used with `if` followed by a logical expression in order to apply the command to a subset of observations. The logical operators `==` and `~=` stand for "equal

to" and "not equal to", < and <= for "less than" and "less than or equal to" and > and >= for "greater than" and "greater than or equal to". The characters ~, & and | represent "not", "and", and " or", respectively. An example of the use of `if` is

```
gen x = y + z if t==1
```

which sets `x` equal to `y + z` for all observations where `t` equals 1 and to missing otherwise.

Logical expressions evaluate to 1 (true) or 0 (false) and can be used to create dummy variables:

```
gen x = y == 2
```

One important thing to keep in mind when using logical expressions is that missing values (represented by a dot) are interpreted as very large numbers. The command

```
gen x = y >= 2
```

would result in `x` being equal to 1 when `y` is greater or equal to 2 or missing!

A very convenient way of creating dummy variables is to use the `tabulate` command:

```
tab item, gen(i)
```

This creates variables `i1`, `i2`, etc. which are dummy variables for the first, second, etc. largest values of `item`, respectively. Within an estimation command, we can generate dummy variables using the `xi:` syntax.

```
xi: regress y i.item
```

Here the dummy variables `_Iitem_2`, `_Iitem_3`, etc. are generated because, in the regression command, the explanatory variable, `item`, is preceded by `i.`. These dummy variables are used as explanatory variables in the regression. The suffix of a given dummy variable corresponds to the value of `item` for which it is an indicator. No dummy variable is created for the lowest value of `item`. (In Stata 6, the dummy variables are called `Iitem_2`, `Iitem_3`, etc.)

Stata stores the results of most commands. For example, after `summarize`, we can access the mean using the expression `r(mean)`, e.g.

```
summ x
display r(mean)
```

After estimation commands, we can also access many stored results. For example, we can use `e(b)` to get the vector (actually a matrix with one row) of parameter estimates and `e(V)` to get the covariance matrix of the parameter estimates. Run a regression followed by the commands

```
matrix a=e(b)
matrix list a
matrix v=e(V)
matrix list v
```

The correlation matrix of the estimated parameters can be obtained from the covariance matrix using

```
matrix c=corr(v)
```

Look up the relevant estimation command in the Reference manual to find out how to access different results.

Very useful post-estimation commands include `testparm`, `test` and `lincom`. Assume that `group` is a categorical variable with values 1,2 or 3 and `y` is some continuous variable. We can use the following commands

```
xi: regress y i.group
testparm _Igroup*
test _Igroup_2=_Igroup_3
lincom _Igroup_2-_Igroup_3
```

to, respectively, run the regression, test the null hypothesis that the coefficients of both dummy variables are 0 (i.e. there are no group difference in mean), test that the coefficients of the dummy variables for groups 2 and 3 are the same (no difference in means between groups 2 and 3) and form a 95% confidence interval for the difference between the coefficients of dummy variables 2 and 3 (for the difference in means between groups 2 and 3).

In order to use ***gllamm****, all responses need to be stacked into a single response vector.* For example, if we have measurement occasions j for subjects i, this may be viewed as a multivariate dataset in which each occasion j is represented by a variable `resultj` and the subject identifier is in the variable `ind`. However, for `gllamm`, we need one single, long, response vector containing the responses for all occasions for all subjects, as well as two variables `ind` and `t` to represent the indices i and j, respectively. The two "data shapes" are called wide and long, respectively. We start from the wide shape with variables `ind`, `result1` and `result2`:

```
. list
          ind    result1    result2
  1.        1          0          0
  2.        2          0          1
  3.        3          0          1
```

and convert this to the long shape with variables `result`, `t`, and `ind` using `reshape`:

```
. reshape long result, i(ind) j(t)
(note: j = 1 2)

Data                               wide   ->   long
-----------------------------------------------------------------------------
Number of obs.                        3   ->       6
Number of variables                   3   ->       3
j variable (2 values)                     ->   t
xij variables:
                         result1 result2   ->   result
-----------------------------------------------------------------------------
```

Giving:

```
. list
```

```
          ind          t     result
  1.        1          1          0
  2.        1          2          0
  3.        2          1          0
  4.        2          2          1
  5.        3          1          0
  6.        3          2          1
```

(We could convert the data back to wide shape using

```
reshape wide result, i(ind) j(t)
```

but we will not)

This dataset becomes like the one used in the explanation of the `weight()` option in the syntax for `gllamm` if we define `occ`, an identifier for the rows in the dataset. Here we can use `generate`, abbreviated `gen`, together with Stata's running observation index `_n`:

```
. gen occ=_n
. list ind occ result

          ind        occ     result
  1.        1          1          0
  2.        1          2          0
  3.        2          3          0
  4.        2          4          1
  5.        3          5          0
  6.        3          6          1
```

If we did not already have `t`, an index for occasions within individuals (equal to 1,2 for each individual), we could create one using `by varlist:` as follows:

```
. sort ind t
. by ind: gen j=_n
. list ind occ result j

          ind        occ     result          j
  1.        1          1          0          1
  2.        1          2          0          2
  3.        2          3          0          1
  4.        2          4          1          2
  5.        3          5          0          1
  6.        3          6          1          2
```

Here, the data are first sorted in ascending order of `ind` and within groups having the same value of `ind`, in ascending order of `t`. `by varlist:` is then used to repeat the same command for each combination of values of `varlist`, i.e. for each value of `ind` in this case. (The data must be sorted by `ind` for this to work.) A very useful feature of `by varlist:` is that it causes the observation index `_n` to count from 1 within each of the groups defined by the unique combinations of the values of `varlist`. The macro `_N` represents the total number of observations, but when used with `by varlist:`, it represents the number of observations within the groups. For example,

```
sort ind result
by ind: list result if _n==_N
```

lists the largest value of `result` for each value of `ind`.

We now collapse the data using the `collapse` command to form level 1 weights and list the data:

```
. gen cons=1
. collapse (sum) wt1=cons, by(ind result)
. gen occpat = _n
. list ind result occpat wt1

          ind     result     occpat        wt1
  1.        1          0          1          2
  2.        2          0          2          1
  3.        2          1          3          1
  4.        3          0          4          1
  5.        3          1          5          1
```

In the `collapse` command, the list of variables specified in the `by()` option determines what groups of observations will be represented by a single line of data in the collapsed dataset. Here, a row of data has been created for each unique combination of values of `ind` and `result` within `ind`. The aggregated variable here is be called `wt1` and is equal to the sum (specified in brackets) of `cons`. (The syntax for forming the mean of `occ` would be `(mean) mnocc=occ`.)

Here we could also form level 2 weights. However, in practice it is safest to create level 2 weights (when the data are in wide form) followed by level 1 weights when the data are in long form. We suggest the following exercise to the reader: Enter the data

```
          ind    result1    result2
  1.        1          0          0
  2.        2          0          1
  3.        3          0          1
```

into Stata. Form level 2 weights, then reshape to long and form level 1 weights. You should end up with dataset C in the `weight()` entry of the `gllamm` syntax.

Another command we will make use of is `expand`. The command

```
expand 3
```

replaces each line of data with three replicates of itself. If the multilevel data are stored in separate files, *file1.dta* for the level 1 variables, *file2.dta* for the level 2 variables, etc., Stata's `merge` command can be used to merge the data as required by the `gllamm` command. Assume `subj` is the level 1 identifier, `class` is the level 2 identifier and `school` is the level 3 identifier. First sort the files :

```
use file2, clear
sort school class
save file2, replace
use file3, clear
sort school
save file3, replace
```

Then read the level 1 file and merge in the others:

```
use file1, clear
sort school class
merge school class using file2
drop _merge
sort school
merge school using file3
```

The `merge` command automatically expands the class level data before adding it to the individual level data, etc.

References

Albert, P. S., & Follmann, D. A. 2000. Modeling repeated count data subject to informative dropout. *Biometrics*, **56**, 667–677.

Bartholomew, D. J., & Knott, M. 1999. *Latent Variable Models and Factor Analysis.* London: Arnold.

Bock, R. D., & Lieberman, M. 1970. Fitting a response model for n dichotomously scored items. *Psychometrika*, **33**, 179–197.

Bollen, K. A. 1989. *Structural Equations with Latent Variables.* New York: Wiley.

Breslow, N. E., & Clayton, D. G. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, **88**, 9–25.

Breslow, N. E., & Lin, X. 1995. Bias correction in generalised linear mixed models with a single component of overdispersion. *Biometrika*, **82**, 81–91.

Bryk, A. S., & Raudenbush, S. W. 1992. *Hierarchical Linear Models, Applications and Data Analysis Methods.* Newbury Park, CA: Sage Publications.

Bryk, A. S., Raudenbush, S. W., & Congdon, R.T. 1996. *HLM. Hierarchical Linear and Nonlinear Modeling with the HLM/2L and HLM/3L Programs.* Chicago: Scientific Software International.

Clayton, D. 1988. The analysis of event history data: a review of progress and outstanding problems. *Statistics in Medicine*, **7**, 819–841.

Croon, M. A. 1989. Latent class models for the analysis of rankings. *Pages 99–121 of:* De Soete, G., & Klauer, K. C. (eds), *New Developments in Psychological Choice Modeling.* North-Holland: Elsevier.

Crouch, E. A. C., & Spiegelman, D. 1990. The evaluation of integrals of the form $\int f(t)exp(-t^2)dt$: Application to logistic-normal models. *Journal of the American Statistical Association*, **85**, 464–469.

Davies, R. 1987. Mass point methods for dealing with nuisance parameters in longitudinal studies. *Pages 88–109 of:* Crouchley, R. (ed), *Longitudinal Data Analysis.* Aldershot: Averbury.

Davies, R., & Pickles, P. 1987. A joint trip timing store-type choice model for grocery shopping, including inventory effects and nonparametric control for omitted variables. *Transportation Research*, **21A**, 345–361.

Dohoo, I.R., Tillard, E., Stryhn, H., & Faye, B. 2001. The use of multilevel models to evaluate sources of variation in reproductive performance in dairy cattle. *Preventative Veterinary Medicine*, in press.

Dunn, G., Everitt, B., & Pickles, A. 1993. *Modelling Covariances and Latent Variables using EQS.* London: Chapman & Hall.

Follmann, D. A., & Lambert, D. 1989. Generalizing logistic regression by nonparametric mixing. *Journal of the American Statistical Association*, **84**, 295–300.

Goldstein, H. 1995. *Multilevel Statistical Models.* London: Arnold.

Goldstein, H., Rasbash, J., Plewis, I., Draper, D., Browne, W., Yang, M., Woodhouse, G., & Healy, M. 1998. *A User's Guide to MLwiN.* London: Multilevel Models Project, Institute of Education, University of London.

Gould, W., & Sribney, W. 1999. *Maximum Likelihood Estimation with Stata.* College Station, TX: Stata Press.

Heckman, J., & Singer, B. 1984. A method of minimising the impact of distributional assumptions in econometric models for duration data. *Econometrica*, **52**, 271–320.

Hu, P., Tsiatis, A. A., & Davidian, M. 1998. Estimating the parameters in the Cox model when the covariate variables are measured with error. *Biometrics*, **54**, 1407–1419.

Kreft, I., & De Leeuw, J. 1998. *Introducing Multilevel Modeling.* London: Sage Publications.

Leese, M. N., White, I. R., Schene, A.H., Koeter, M.W.J., Ruggeri, M., Gaite, L., & the EPSILON Study Group. 2001. Reliability in Multi-Site Psychiatric Studies. *International Journal of Methods in Psychiatric Research*, **10**, in press.

Lesaffre, E., & Spiessens, B. 2001. On the effect of the number of quadrature points in a logistic random-effects model: an example. *Applied Statistics*, **50**, 325–335.

Lin, X., & Breslow, N. E. 1995. Analysis of correlated binomial data in logistic-normal models. *Journal of Statistical Computing and Simulation*, **55**, 133–146.

Lindsay, B. G., Clogg, C. C., & Grego, J. 1991. Semiparametric estimation in the Rasch model and related exponential response models, including a simple latent class model for item analysis. *Journal of the American Statistical Association*, **86**, 96–107.

Little, R. J. A., & Rubin, D. B. 1987. *Statistical Analysis with Missing Data.* New York: Wiley.

Liu, Q., & Pierce, D. A. 1994. A note on Gauss-Hermite quadrature. *Biometrika*, **81**, 624–629.

Longford, N. T. 1993. *Random Coefficient Models.* Oxford: Oxford University Press.

Luce, R. D. 1959. *Individual Choice Behavior.* New York: Wiley.

Magder, S.M., & Zeger, S. L. 1996. A smooth nonparamtric estimate of a mixing distribution using mixtures of Gaussians. *Journal of the American Statistical Association*, **11**, 86–94.

Maughan, B., Pickles, A., Rowe, A., Costello, R., & Angold, A. 2000. Developmental trajectories of aggressive and non-aggressive conduct problems. *Journal of Quantitative Criminology*, **16**, 199–221.

McCullagh, P., & Nelder, J. A. 1989. *Generalized Linear Models.* London: Chapman & Hall.

McCulloch, C. E., & Searle, R. S. 2001. *Generalized, Linear and Mixed Models.* New York: Wiley.

McFadden, D. 1973. Conditional logit analysis of qualitative choice behavior. *Pages 105–142 of:* Zarembka, P. (ed), *Frontiers in Econometrics*. New York: Academic Press.

Morris, J. N., Marr, J. W., & Clayton, D. G. 1977. Diet and heart: postscript. *British Medical Journal*, **2**, 1307–1314.

Muthén, L. K., & Muthén, B. O. 1998. *Mplus User's Guide*. Los Angeles, CA: Muthén & Muthén.

Naylor, J. C., & Smith, A. F. M. 1982. Applications of a method for the efficient computation of posterior distributions. *Applied Statistics*, **31**, 214–225.

Pickles, A., & Crouchley, R. 1994. Generalizations and applications of frailty models for survival and event data. *Statistics in Medicine*, **3**, 263–278.

Pickles, A., & Crouchley, R. 1995. A comparison of frailty models for multivariate survival data. *Statistics in Medicine*, **14**, 1447–1461.

Rabe-Hesketh, S., & Everitt, B. S. 2000. *Handbook of Statistical Analyses using Stata (2nd Edition)*. Boca Raton: Chapman & Hall/CRC.

Rabe-Hesketh, S., & Pickles, A. 1999. Generalised linear latent and mixed models. *Pages 332–339 of:* Friedl, H., Berghold, A., & Kauermann, G. (eds), *14th International Workshop on Statistical Modeling*.

Rabe-Hesketh, S., & Pickles, A. 2001. Correcting for measurement error using a nonparametric exposure distribution and non-normal errors. *Submitted*.

Rabe-Hesketh, S., & Skrondal, A. 2001. Parameterisation of multivariate random effects models for categorical data. *Biometrics*, in press.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. 2001a. Generalized multilevel structural equation modelling. *Submitted*.

Rabe-Hesketh, S., Pickles, A., & Skrondal, A. 2001b. GLLAMM: A general class of multilevel models and a Stata program. *Multilevel Modelling Newsletter*, **13**, 17–23.

Rabe-Hesketh, S., Touloupulou, T., & Murray, R. M. 2001c. Multilevel modeling of cognitive function in schizophrenics and their first degree relatives. *Multivariate Behavioral Research*, **36**, 279–298.

Rabe-Hesketh, S., Yang, S., & Pickles, A. 2001d. Multilevel models for censored and latent responses. *Statistical Methods in Medical Research*, in press.

Rabe-Hesketh, S., Pickles, A., & Skrondal, A. 2001e. Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal*, **submitted**, 0.

Rodriguez, B., & Goldman, N. 1995. An assessment of estimation procedures for multilevel models with binary responses. *Journal of the Royal Statistical Society, Series A*, **158**, 73–89.

Skrondal, A. 1996. *Latent trait, multilevel and repeated measurement modelling with incomplete data of mixed measurement levels*. Oslo: Section of Medical Statistics, University of Oslo.

Skrondal, A., & Rabe-Hesketh, S. 2001a. Analysis of clustered survival data with covariate measurement error. *In preparation*, **0**, ?

Skrondal, A., & Rabe-Hesketh, S. 2001b. Multilevel logistic regression for polytomous data and rankings. *Submitted.*

Snijders, T. A. B., & Bosker, R. J. 1999. *Multilevel Analysis.* London: Sage Publications.

Stata Manuals 1-4. 2001. *Stata Reference Manuals.* College Station, TX.

StataCorp. 2001. *Stata Statistical Software: Release 7.* College Station, TX.

Stryhn, H., Dohoo, I.R., Tillard, E., & Hagedorn-Olsen, T. 2000. Simulation as a tool of validation in hierarchical generalized linear models. *Pages 1136–1138 of: International Symposium on Veterinary Epidemiology and Economics.*

Woodhouse, G. (ed). 1995. *A guide to MLn for new users.* London: Multilevel Models Project, Institute of Education, University of London.

Yang, M., Rasbash, J., Goldstein, H., & Barbosa, M. 1999. *MLwiN macros for advanced multilevel modelling.* University of Education, London: Multilevel Models Project.

Index