University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2004 *Paper* 160

GLLAMM Manual

Sophia Rabe-Hesketh[*] Anders Skrondal[†]

Andrew Pickles[‡]

[*]Graduate School of Education and Graduate Group in Biostatistics, UC Berkeley, sophiarh@berkeley.edu

[†]Biostatistics Group, Division of Epidemiology, Norwegian Institute of Public Health, Oslo, Norway

[‡]School of Epidemiology and Health Science and CCSR, The University of Manchester , England

This working paper is hosted by The Berkeley Electronic Press (bepress) and may not be commercially reproduced without the permission of the copyright holder.

http://biostats.bepress.com/ucbbiostat/paper160

Copyright ©2004 by the authors.

GLLAMM Manual

Sophia Rabe-Hesketh, Anders Skrondal, and Andrew Pickles

Abstract

This manual describes a Stata program gllamm that can estimate Generalized Linear Latent and Mixed Models (GLLAMMs). 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 (common factors or random effects) can be assumed to be discrete or to have a multivariate normal distribution. Examples of models in this class are multilevel generalized linear models or generalized linear mixed models, multilevel factor or latent trait models, item response models, latent class models and multilevel structural equation models. The program can be downloaded from http://www.gllamm.org.

Introduction and Disclaimer

`gllamm` is a Stata program to fit GLLAMMs (Generalized 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 generalized linear models or generalized linear mixed models, multilevel factor or latent trait models, latent class models and multilevel structural equation models. The program runs in the statistical package Stata (see http://www.stata.com).

This manual replaces the earlier version (Technical Report 2001/01). We have changed the notation to be consistent with our papers and books, added descriptions of some new options, updated the references, and made some corrections. We have also reworked the examples in Stata 8 (which has much more attractive graphics). However, we have not added new examples. This is partly because we are writing a book on the `gllamm` program, including the model framework and applications, to be published by Stata Press. This book will replace the manual and complement the more theoretical account given in

- Skrondal, A. and Rabe-Hesketh, S. (2004). *Generalized Latent Variable Modeling: Multilevel, Longitudinal and Structural Equation Models.* Boca Raton: Chapman & Hall/CRC.

Further examples using `gllamm`, including applications from the above book are available from:

http://www.gllamm.org/examples.html

The program and manual are free. In return, we ask you to refer to `gllamm` if you use it in publications and give one of the citations below:

- For generalized linear mixed models or multilevel regression models and adaptive quadrature:
  - Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2005). Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics*, in press.
  - Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2002). Reliable estimation of generalized linear mixed models using adaptive quadrature. *The Stata Journal* **2** (1), 1-21.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

- For factor, IRT or structural equation models:
  - Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2004a). Generalized multilevel structural equation modelling. *Psychometrika* **69** (2), 167-190.
- For nominal data, discrete choice data and rankings:
  - Skrondal, A. and Rabe-Hesketh, S. (2003b). Multilevel logistic regression for polytomous data and rankings. *Psychometrika* **68** (2), 267-287.
- For nonparametric maximum likelihood estimation (NPMLE) and covariate measurement error models:
  - Rabe-Hesketh, S., Pickles, A. and Skrondal, A. (2003a). Correcting for covariate measurement error in logistic regression using nonparametric maximum likelihood estimation. *Statistical Modelling* **3** (3), 215-232.
  - Rabe-Hesketh, S., Skrondal, A. and Pickles, A. (2003b). Maximum likelihood estimation of generalized linear models with covariate measurement error. *The Stata Journal* **3** (4), 385-410.

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 in the future.

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 sophiarh@berkeley.edu. We would also be grateful if you could let us know of any publications (including 'in press') using `gllamm` so we can add the reference to the list of publications at:

http://www.gllamm.org/pub.html

Thanks to Kit Baum, the program can be obtained from the Statistical Software Components (SSC) archive; use the Stata commands

```
    ssc describe gllamm
    ssc install gllamm, replace
```

and see

http://www.gllamm.org/install.html

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

for more information on installing `gllamm`. Importantly, this web page shows how you can make sure you are actually using the most current version of `gllamm`. The most common reason for errors is due to older versions of any of the `gllamm ado` files being located higher up in the `adopath`.

All sorts of information on `gllamm`, including worked examples for books and papers, information on courses and workshops and an up-to-date list of publications and are available at

http://www.gllamm.org

Note that `gllamm6` (Rabe-Hesketh, Pickles and Taylor, 2000) is a very out-of-date version of the program. 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.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Contents

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 1

Generalized linear latent and mixed models

GLLAMMs (Generalized 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 generalized linear models or generalized linear mixed models, multilevel factor or latent trait models, latent class models and multilevel structural equation models.

We will only give a very brief overview of GLLAMMs here and refer to Skrondal and Rabe-Hesketh (2004) and Rabe-Hesketh, Skrondal and Pickles (2004a) for detailed accounts of GLLAMMs and their applications.

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 $\boldsymbol{\eta}$. As we will see later, *the latent variables can be interpretable as random effects (random intercepts and coefficients) or common factors.*

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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 and the original units as level-2 clusters 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 $\boldsymbol{\eta}$ is specified via a link function $g(\cdot)$ and a linear predictor ν as

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

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

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

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 $\boldsymbol{\beta}$, $\eta_m^{(l)}$ is the mth latent variable at level l and $\boldsymbol{\eta}$ in equation (1.1) is the vector of latent variables,

$$\boldsymbol{\eta} = (\eta_1^{(2)}, \eta_2^{(2)}, \ldots, \eta_{M_2}^{(2)}, \eta_1^{(3)}, \eta_2^{(3)}, \ldots, \eta_{M_3}^{(3)}, \ldots, \eta_1^{(L)}, \eta_2^{(L)}, \ldots, \eta_{M_L}^{(L)}) \tag{1.4}$$

Each latent variable is multiplied by a linear combination of explanatory variables $\mathbf{z}_m^{(l)\prime}\boldsymbol{\lambda}_m^{(l)}$. 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 $\boldsymbol{\eta}$. 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 generalized linear models (or generalized linear mixed models) and multilevel factor models as special cases.

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

1.1.1 Multilevel generalized linear models

There are a large number of books on multilevel models, see for example Skrondal and Rabe-Hesketh (2004), Raudenbush and Bryk (2002), Snijders and Bosker (1999), Longford (1993), Goldstein (2003) and McCulloch and Searle (2001).

To write down a multilevel generalized linear model, or generalized 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 generalized linear models are therefore a special case of GLLAMMs with

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

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

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

$$\nu_{ijk} = \mathbf{x}'_{ijk}\boldsymbol{\beta} + \eta_{1jk}^{(2)} + \eta_{2jk}^{(2)} z_{2ijk}^{(2)} + \eta_{1k}^{(3)}. \tag{1.6}$$

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

Multilevel generalized 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 (e.g. Raudenbush and Bryk, 2002). These coefficients are then regressed on level-2 covariates and have level 2 residuals, etc. For a two-level linear random coefficient model (using subscripts i and j for levels 1 and 2), the relationship between the response variable and the level 1 covariates has the form

$$y_{ij} \;=\; \eta_{0j} + \eta_{1j}x_{ij} + \epsilon_{ij}, \tag{1.7}$$

where x_{ij} is a unit-level covariate and η_{0j} and η_{1j} are the intercept and slope for the jth cluster, respectively. This is sometimes called the level-1 model. The between-cluster variability of the intercept and slope is modeled using level-2 models,

$$\begin{aligned} \eta_{0j} &= \gamma_{00} + \gamma_{01}w_j + \zeta_{0j}, \\ \eta_{1j} &= \gamma_{10} + \gamma_{11}w_j + \zeta_{1j}. \end{aligned} \tag{1.8}$$

Substituting (1.8) for the coefficients η_{0j} and η_{1j} in the level-1 model (1.7), we obtain the reduced form

$$\begin{aligned} y_{ij} &= \underbrace{\gamma_{00} + \gamma_{01}w_j + \zeta_{0j}}_{\eta_{0j}} + \underbrace{(\gamma_{10} + \gamma_{11}w_j + \zeta_{1j})}_{\eta_{1j}} x_{ij} + \epsilon_{ij} \\ &= \gamma_{00} + \gamma_{01}w_j + \gamma_{10}x_{ij} + \gamma_{11}(w_j x_{ij}) + \zeta_{0j} + \zeta_{1j}x_{ij} + \epsilon_{ij}. \end{aligned}$$

We have only used one explanatory variable $z_{m1}^{(2)}$ 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 i are the measurement occasions of a longitudinal study

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

and the level-2 units j 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 specifying

$$\nu_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \eta_j^{(2)}(z_{bij} + \lambda_g z_{gij}) \tag{1.9}$$

so that the random intercept variance is $\mathrm{Var}(\eta_j^{(2)})$ for boys and $\lambda_g^2\mathrm{Var}(\eta_j^{(2)})$ 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 depend on covariates (see Section 1.2). Models with discrete random effects, such as latent trajectory models, are discussed in Section 5.2.

Examples of multilevel generalized linear models are given in Chapter 3. Papers using `gllamm` for such models include Rabe-Hesketh *et al.* (2001b), Leese *et al.* (2001) and Marshall *et al.* (2004) in psychiatry, Dohoo *et al.* (2001), Stryhn *et al.* (2000) and Vigre *et al.* (2004) in veterinary medicine, Kaufman *et al.* (2003) and Skrondal and Rabe-Hesketh (2003) in epidemiology, Ebel *et al.* (2003), Vincent *et al.* (2003), Glance *et al.* (2003), Panageas *et al.* (2003), Daucourt *et al.* (2003) and others in other areas of medicine, Boylan (2004) in law, Holmås (2002) in economics and Grilli and Rampichini (2003) and Pagani and Seghieri (2002) in education.

1.1.2 Multilevel factor and item response models

For a treatment of factor models for continuous and discrete responses, see Skrondal and Rabe-Hesketh (2004) and Bartholomew and Knott (1999). Multilevel factor models for continuous data are discussed in Longford (1993) and for discrete data in Rabe-Hesketh, Skrondal and Pickles (2004).

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 I variables $i = 1, \ldots, I$ observed on each subject j and stack the variables into a single response vector indexed ij. This is shown in the table below:

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

A single level unidimensional factor model can be written as

$$\begin{aligned}\nu_{ij} &= \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \eta_{1j}^{(2)}(\lambda_{11}^{(2)} z_{11i}^{(2)} + \lambda_{12}^{(2)} z_{12i}^{(2)} + \lambda_{13}^{(2)} z_{13i}^{(2)} + \ldots)\\ &= \beta_i + \eta_{1j}^{(2)}\lambda_{1i}^{(2)},\end{aligned} \tag{1.10}$$

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

$$z^{(2)}_{1pi} = \begin{cases} 1 & \text{if p = i} \\ 0 & \text{otherwise} \end{cases} \tag{1.11}$$

In equation (1.10), β_i is the intercept, $\eta^{(2)}_j$ is the common factor and $\lambda^{(2)}_{1i}$ is the factor loading for the ith 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 $y_{ij} = \nu_{ij} + \epsilon_{ij}$. In factor models for non-normal responses, the level 1 variability is implicit in the distribution family of the chosen generalized linear model (see Section 1.2).

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

$$\begin{aligned} \nu_{ij} &= \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \eta^{(2)}_{1j}(\lambda^{(2)}_{11} z^{(2)}_{11i} + \lambda^{(2)}_{12} z^{(2)}_{12i} + \ldots) + \eta^{(2)}_{2j}(\lambda^{(2)}_{21} z^{(2)}_{21i} + \lambda^{(2)}_{22} z^{(2)}_{22i} + \ldots) \\ &= \beta_i + \eta^{(2)}_{1j}\lambda^{(2)}_{1i} + \eta^{(2)}_{2j}\lambda^{(2)}_{2i}. \end{aligned}$$

The purpose of the dummy variables $z^{(2)}_{1pi}$ and $z^{(2)}_{2pi}$ is to pick out the correct factor loading for the items. Typically, not all items load on all factors so that only some of the dummies are needed. 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

$$\begin{aligned} \nu_{ijk} &= \beta_1 x_{1ijk} + \beta_2 x_{2ijk} + \ldots + \eta^{(2)}_{1jk}(\lambda^{(2)}_{11} z^{(2)}_{11i} + \lambda^{(2)}_{12} z^{(2)}_{12i} + \ldots) \\ &\quad + \eta^{(3)}_{(I+1)k}(\lambda^{(3)}_{11} z^{(3)}_{11i} + \lambda^{(3)}_{12} z^{(3)}_{12i} + \ldots) + \eta^{(3)}_{1k} z^{(3)}_{11i} + \eta^{(3)}_{2k} z^{(3)}_{12i} + \ldots \\ &= \beta_i + \eta^{(2)}_{1jk}\lambda^{(2)}_{1i} + \eta^{(3)}_{(I+1)k}\lambda^{(3)}_{1i} + \eta^{(3)}_{ik}, \end{aligned} \tag{1.12}$$

where $z^{(2)}_{1pi} = z^{(3)}_{1pi}$ are dummy variables for the items as in equation (1.11), $\eta^{(2)}_{1jk}$ is the lower level common factor, $\eta^{(3)}_{(I+1)k}$ is the higher level common factor and $\eta^{(3)}_{1k}$, $\eta^{(3)}_{2k}$, 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 modeled 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. Rabe-Hesketh, Skrondal and Pickles (2004b) discuss multilevel item response models. Other applications of measurement models using `gllamm` include Campbell *et al.* (2001), Baker and Hann (2001) and Finkelstein (2002) in medicine and Hardouin and Mesbah (2004) in education.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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. This is accomplished by specifying a model for the log of the scale parameter

$$\ln \sigma_{ijk} \;=\; \mathbf{z}_{ijk}^{(0)\prime}\boldsymbol{\alpha}. \tag{1.13}$$

The available links and families allow many different response processes to be modeled 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 modeled, for example logistic regression with measurement errors in a continuous covariate. Mixing of the identity link and Gaussian family with the probit link and binomial family allows censored normally distributed responses to be modeled (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.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Leese *et al.* (2001) use `gllamm` to specify heteroscedasticity at levels 1 and 2. Rabe-Hesketh and Pickles (1999) and Rabe-Hesketh, Pickles and Skrondal (2003a) mix the identity and logit links to estimate logistic regression models with covariate measurement error. The wrapper `cme` (Rabe-Hesketh, Skrondal and Pickles) (2003b) makes it easy to estimate some covariate measurement error models in `gllamm`. Rabe-Hesketh *et al.* (2001c), Holmås (2002) and Arulampalam (2004) analyze different types of discrete time survival models and Skrondal and Rabe-Hesketh (2003b) consider nominal data such as polytomous responses or discrete choice and rankings.

1.3 Structural equations for the latent variables

Books on structural equation models include Skrondal and Rabe-Hesketh (2004), Dunn, Everitt and Pickles (1993) and Bollen (1989). The models discussed so far can be viewed as response models specifying the relationship between observed responses (or indicators) and latent and observed explanatory variables. In addition, we can specify regressions of latent continuous or discrete variables on explanatory variables as well as relationships among latent continuous variables.

1.3.1 Continuous latent variables

Relationships among latent continuous variables can be written as a matrix equation for the vector of latent variables $\boldsymbol{\eta}$ whose M elements are the latent variables varying at levels 2 to L,

$$\boldsymbol{\eta} = (\eta_1^{(2)}, \eta_2^{(2)}, \ldots, \eta_{M_2}^{(2)}, \eta_1^{(3)}, \eta_2^{(3)}, \ldots, \eta_{M_3}^{(3)} \ldots \eta_1^{(L)}, \eta_2^{(L)}, \ldots, \eta_{M_L}^{(L)}). \tag{1.14}$$

(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

$$\boldsymbol{\eta} = \mathbf{B}\boldsymbol{\eta} + \boldsymbol{\Gamma}\mathbf{w} + \boldsymbol{\zeta} \tag{1.15}$$

where $\mathbf{B}$ is an $M \times M$ upper diagonal regression matrix, $\mathbf{w}$ is a vector of q covariates, $\boldsymbol{\Gamma}$ is an $M \times q$ regression 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 $\boldsymbol{\eta}$.

The regressions of latent variables on other latent variables must be such that the elements of the $\boldsymbol{\eta}$ 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 $\boldsymbol{\eta}$ 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 $\boldsymbol{\eta}$ 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.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

MIMIC models

MIMIC (Multiple Indicator Multiple Cause) models are factor models (where the responses are measures or '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 (1.15), we only require

$$\boldsymbol{\eta} = \boldsymbol{\Gamma}\mathbf{w} + \boldsymbol{\zeta}. \tag{1.16}$$

The covariate measurement error models described in Section 6.1 can be viewed as MIMIC models with and without direct effects. Section 8.4 describes a MIMIC model with ordinal indicators.

General multilevel structural equation models

Combining (1.15) 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, Skrondal and Pickles (2004a). 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.

1.3.2 Discrete latent variables

For discrete latent variables, the structural model is the model for the (prior) probabilities that the units belong to the corresponding latent classes. For a level-2 unit j, let the probability of belonging to class c be denoted as π_{jc},

$$\pi_{jc} \;\equiv\; \Pr(\boldsymbol{\eta}_j^{(2)} = \mathbf{e}_c).$$

This probability may depend on covariates $\mathbf{v}_j$ through a multinomial logit model

$$\pi_{jc} \;=\; \frac{\exp(\mathbf{v}_j'\boldsymbol{\varrho}_c)}{\sum_d \exp(\mathbf{v}_j'\boldsymbol{\varrho}_d)}, \tag{1.17}$$

where $\boldsymbol{\varrho}_c$ are regression parameters with $\boldsymbol{\varrho}_1 = 0$ imposed for identification. Such a 'concomitant variable' latent class model is used for instance by Dayton and MacReady (1988) and Formann (1992). The multinomial logit parametrization is useful even if the class membership does not depend on covariates since it forces latent class probabilities to sum to one.

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).

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

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 $\boldsymbol{\eta}$ directly. The latent variables at a level l may be assumed to have a

- multivariate normal distribution with zero mean and covariance matrix $\boldsymbol{\Sigma}_l$ 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 characteristic represented by the latent variable, e.g. in their intercepts.

If the number of points, or masses, is chosen to achieve the largest possible likelihood, the nonparametric maximum likelihood estimator (NPML) can be achieved (Lindsay *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 greatest 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, Pickles and Skrondal (2003a) and Holmås (2002) on nonparametric maximum likelihood estimation and Rabe-Hesketh and Pickles (1999) on a latent transition model.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 2

The `gllamm` program

The `gllamm` program runs within Stata 6, 7 and 8 (StataCorp 2003) 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 and other predictions. `gllasim` can be used to simulate from the model.

2.1 Implementation

`gllamm` uses Stata's `ml` with method `d0` to maximize the likelihood. (See the Stata reference manuals under `ml` and `maximize`; for details of the modified Newton Raphson algorithm, see also Gould, Pitblado and Sribney (2003)). For a two-level model, the likelihood is given by

$$\prod_j \int \{\prod_i f(y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\eta}_j)\} g(\boldsymbol{\eta}_j)\, \mathrm{d}\boldsymbol{\eta}_j \tag{2.1}$$

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

$$\prod_j \sum_r \pi_r \prod_i f(y_{ij}|\mathbf{x}_{ij}, \boldsymbol{\eta}_j = \mathbf{e}_r) \tag{2.2}$$

where the locations $\mathbf{e}_r$ 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 $\mathbf{v}$, $\boldsymbol{\eta} = \mathbf{L}\mathbf{v}$ where $\mathbf{L}$ is a lower triangular matrix. The multiple integral is then approximated by evaluating 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 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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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, Skrondal and Pickles (2002, 2005) for details.

In (2.1), the contribution to the likelihood from the jth 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 a L-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 positive semi-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

$$\mathrm{Var}(\eta_m) = \sum_r \widehat{\pi}_r \widehat{e}^2_{rm} \tag{2.3}$$

and the covariances are estimated as

$$\mathrm{Cov}(\eta_m, \eta_{m'}) = \sum_r \widehat{\pi}_r \widehat{e}_{rm} \widehat{e}_{rm'} \tag{2.4}$$

where $\widehat{\pi}_r$ and $\widehat{e}_m$ 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 $\widehat{e}_{rm}$ by $\widehat{e}_{rm} - \sum_r \widehat{\pi}_r \widehat{e}_{rm}$.

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 test statistics will no longer have conventional chi-square distributions under the null.

Linear constraints can be specified for the transformed versions of the parameters that are used during maximization. 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`. For a single continuous random

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

effect in a two-level model, the posterior mean is obtained using

$$\widetilde{\eta}_j = \frac{\int \eta_j g(\eta_j) \prod_i f(y_{ij}|\mathbf{x}_{ij}, \eta_j)\, \mathrm{d}\eta_j}{\int g(\eta_j) \prod_i f(y_{ij}|\mathbf{x}_{ij}, \eta_j)\, \mathrm{d}\eta_j}. \tag{2.5}$$

If there is instead a single discrete random effect, the posterior mean is obtained using

$$\widetilde{\eta}_j = \sum_r \widehat{e}_r \widehat{w}_r \tag{2.6}$$

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

$$\widehat{w}_r = \frac{\widehat{\pi}_r \prod_i f(y_{ij}|\mathbf{x}_{ij}, \eta_j = \widehat{e}_r)}{\sum_r \widehat{\pi}_r \prod_i f(y_{ij}|\mathbf{x}_{ij}, \eta_j = \widehat{e}_r)} \tag{2.7}$$

2.2 Installing gllamm

The easiest way of installing the current version of the program is

```
ssc describe gllamm
ssc install gllamm, replace
```

to use the net commands from within Stata: Alternatively, `net install` from the GLLAMM web page using:

```
net from http://www.gllamm.org
net describe gllamm
net install gllamm, replace
net get gllamm, replace
```

Where the last command is optional; it downloads the auxiliary files.

Another possibility is to store the files *gllamm.ado*, *gllam_ll.ado*, *remcor.ado*, *gllas_yu.ado*, *gllarob.ado*, *gllamm.hlp*, *gllapred.ado gllapred.hlp*, *gllasim.ado*, *gllasim.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.

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.* (2001a).

2.3 Running gllamm

`gllamm` runs in Stata 6 to 8 (StataCorp, 2003). 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

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

Hosted by The Berkeley Electronic Press

brief introduction to Stata. Sections 2.3.1 and 2.3.2 describe the syntax of `gllamm`, `gllapred` and `gllasim` 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 maximize it. Roughly, execution time is proportional to the number of observations and the square of the number of parameters. For quadrature, the time is approximately proportional to the product of the number of quadrature points for all latent variables used. For example, if there are two random effects at level 2 (a random intercept and slope) and 8 quadrature points are used for each random effect, the time will be approximately proportional to 64. Therefore, using 4 quadrature points for each random effect will take only about a quarter (64/16) as long as using 8. For (2-level) discrete latent variables, the time is proportional to the number of points, but the increase in the number of parameters must be taken into account.

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 2.3.4). 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 depvar [varlist] [if exp] [in range] , i(varlist) [ noconstant offset(varname)
   nrf(#,...,#) eqs(eqnames) frload(#,...,#) ip(string) nip(#,...,#)
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

`peqs(`*eqname*`) bmatrix(`*matrix*`) geqs(`*eqnames*`) nocorrel constraints(`*clist*`)` `weight(`*varname*`) pweight(`*varname*`) family(`*familynames*`) fv(`*varname*`)` `denom(`*varname*`) s(`*eqname*`) link(`*links*`) lv(`*varname*`) expanded(`*varname varname string*`) basecategory(`#`) composite(`*varname varname*`...) thresh(`*eqnames*`)` `ethresh(`*eqnames*`) from(`*matrix*`) copy skip long lf0(`# #`) gateaux(`# # #`)` `search(`#`) noest eval init iterate(`#`) adoonly adapt robust cluster(`*varname*`)` `level(`#`) eform allc trace nolog nodisplay dots`]

where the available families and links are:

families	links
`gaussian`	`identity`
`poisson`	`log`
`gamma`	`reciprocal`
`binomial`	`logit`
	`probit`
	`cll` (complementary 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)`.

`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).

`nrf(`#,...,#`)` specifies the number of latent variables M_l at each level l, 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 $\mathbf{z}_m^{(l)\prime}\boldsymbol{\lambda}_m^{(l)}$ 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 unless the `frload()` option is specified. The other coefficients are estimated together with the (co)variance(s) of the random effect(s)

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

`frload`(#,...,#) lists the latent variables for which the first factor loading should be freely estimated along with the other factor loadings. It is up to the user to define appropriate constraints to identify the model. Here the latent variables are referred to as 1 2 3, etc., in the order in which they are defined by the `eqs()` option.

`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, only `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 all `nip` masses freely – the user must then make sure that the mean is not modeled 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.

`peqs`(*eqname*) can be used with the `ip(f)` or `ip(fn)` options to allow the (prior) latent class probabilities to depend on covariates as shown in (1.17). The model for the latent class probabilities is a multinomial logit model with the last latent class as reference category. A constant is automatically included in addition to the covariates specified in the equation command.

`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.

`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 modeled as multivariate normal.

`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.

`weight`(*wt*) specifies that variables `wt1`, `wt2`, etc., contain frequency weights. The suffixes in the variable names determine at what level each weight applies. (If only some of the weight variables exist, e.g. only level 2 weights, the other weights are assumed to be equal to 1.) 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:

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

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.

`pweight`(*varname*) specifies that variables varname1, varname2, etc. contain sampling weights for levels 1, 2, etc. As far as the estimates and log-likelihood are concerned, the effect of specifying these weights is the same as for frequency weights, but the standard errors will be different. Robust standard errors will automatically be provided. This should be used with caution if the sampling weights apply to units at a lower level than the highest level in the multilevel model. The weights are not rescaled; scaling is the responsibility of the user.

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 several families are specified in the `family()` option. The variable indicates which family applies to which observation. A value of one refers to the first family specified in `family()`, etc.

Collection of Biostatistics Research Archive

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

`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 modeled 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.

`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`.

`composite`(*varname varname* ...) specifies that a composite link is used. The first variable is a cluster identifier ("cluster" below) so that linear predictors within the cluster can be combined into a single composite link. The second variable ("ind" below) indicates to which response the composite links defined by the susequent weight variables belong.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Observations with ind=0 have a missing link. The remaining variables (”c1” and ”c2” below) specify weights for the composite links. The composite link based on the first weight variable will go to where ind=1, etc.

```
Example:


Data setup with form of inverse link       Interpretation of
h_i determined by link() and lv():         composite(cluster ind c1 c2)

cluster ind  c1  c2  inverse link           cluster  composite link
  1      1    1  0    h_1                      1       h_1 - h_2
  1      2   -1  1    h_2                      1       n_2 + h_3
  1      0    0  1    h_3             ==>      1       missing
  2      1    1  0    h_4                      2       h_4 + h_5
  2      2    1  1    h_5                      2       h_5 + 2*h_6
  2      0    0  2    h_6                      2       missing
```

`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.

`ethresh`(*eqnames*) is the same as `thresh`(*eqnames*) except that a different parameterization is used for the threshold model. To ensure that $k_{s-1} \leq k_s$, the model is $k_s = k_{s-1} + \exp(xb)$, for response categories $s = 2, \ldots, S$.

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.)

`copy` see `from`(*matrix*)

`skip` see `from`(*matrix*)

`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.

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

Hosted by The Berkeley Electronic Press

`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

`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

`iterate`(#) specifies the (maximum) number of iterations. With the `adapt` option, use of the `iterate(#)` option will cause `gllamm` to skip the "Newton Raphson" iterations usually performed at the end without updating the quadrature locations. `iterate(0)` is like `eval` except that standard errors are computed.

`adoonly` causes `gllamm` to use only ado-code. `gllamm` will be faster if if it uses internalized versions of some of the functions available in Stata 7 (if updated on or after 26 October 2001) and Stata 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.

`robust` specifies that the Huber/White/sandwich estimator of the covariance matrix of the parameter estimates is to be used. If a model has been estimated without the `robust` option, the robust standard errors can be obtained by simply typing `gllamm, robust`.

`cluster`(*varname*) specifies that the highest level units of the GLLAMM model are nested in even higher level clusters where `varname` contains the cluster identifier. Robust standard errors will be provided that take this clustering into account. If a model has been estimated without this option, the robust standard errors for clustered data can be obtained using the command `gllamm, cluster(varname)`.

Collection of Biostatistics Research Archive

`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` displays details of the model being fitted as well as details of the maximum-likelihood iterations.

`nolog` suppresses output for maximum likelihood iterations.

`nodisplay` suppresses output of the estimates but still shows iteration log unless `nolog` is used.

`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* [`if` *exp*] [`in` *range*] [`, u fac p xb ustd cooksd linpred mu marginal us(varname) outcome(#) above(#,...,#) pearson deviance anscombe s ll fsample nooffset adapt adoonly from(matrix)`]

where *varname* is the 'prefix' used for the variables that will contain the predictions and only one of these options may be specified at a time: `xb`, `u`, `fac`, `p`, `linpred`, `mu`, `pearson`, `deviance`, `anscobe`, `s`, `ll`.

Options

`u` causes posterior means and standard deviations of the latent variables to be 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 continuous latent variables, the integration method (ordinary versus adaptive quadrature) and the number of quadrature points used is the same as in the previous call to `gllamm`. If the `gllamm` model includes equations for the latent variables (`geqs` and/or `bmatrix`), the posterior means and standard deviations of the disturbances $\boldsymbol{\zeta}$ are returned.

COBRA
BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

`fac` if the `gllamm` model includes equations for the latent variables (`geqs()` and/or `bmatrix()`), `fac` causes predictions of the latent variables $\boldsymbol{\eta}$ (e.g. factors) to be returned in "varname"m1, "varname"m2, etc. instead of the disturbances $\boldsymbol{\zeta}$. In other words, predictions of the latent variables on the left-hand side of the structural equations are returned.

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

`xb` The fixed effects part of the linear predictor is returned in *varname* including the offset (if there is one) unless the `nooffset` option is used.

`ustd` The standardized posterior means of the disturbances are returned. The sampling standard deviation, namely the square root of the difference between the prior and posterior variances, is only approximate for most models.

`cooksd` returns Cook's distances for the top-level units.

`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.

`mu` returns the expectation of the response, for example the predicted probability in the case of dichotomous responses. By default, the expectation is with respect to the posterior distribution of the latent variables, but see `marginal` and `us()` options. The offset is included (if there is one in the `gllamm` model) unless the `nooffset` option is specified.

`marginal` together with the `mu` option gives the expectation of the response with respect to the prior distribution of the latent variables. This is useful for looking at the 'marginal' or population averaged effects of covariates.

`us`(*varname*) can be used to specify values for the latent variables to calculate conditional quantities, such as the conditional mean of the responses (`mu` option) given the values of the latent variables. Here `varname` specifies the stub-name (prefex) for the variables and `gllapred` will look for "varname"1 "varname"2, etc.

`outcome`(#) specifies the outcome for which the predicted probability should be returned (`mu` option) if there is a nominal response. This option is not necessary if the `expanded()` option was used in `gllamm` since in this case predicted probabilities are returned for all outcomes.

`above`(#,...,#) specifies the events for which the predicted probabilities should be returned (`mu` option) if there are ordinal responses. The probability of a value higher than that specified is returned for each ordinal response. A single number can be given for all ordinal responses (if there are several).

`pearson` returns Pearson residuals. By default, the posterior expectation with respect to the latent variables is returned. The `us()` option can be used to obtain the conditional residual when certain values are substituted for the latent variables.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

`deviance` returns deviance residuals. By default, the posterior expectation with respect to the latent variables is returned. The `us()` option can be used to obtain the conditional residual when certain values are substituted for the latent variables.

`anscombe` returns Anscombe residuals. By default, the posterior expectation with respect to the latent variables is returned. The `us()` option can be used to obtain the conditional residual when certain values are substituted for the latent variables.

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

`ll` returns the log-likelihood contributions of the highest level (level L) units.

`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.

`fsample` causes `gllapred` to return predictions for the full sample (except observations excluded due to the if and in options), not just the estimation sample. The returned log-likelihood may be missing since `gllapred` will not exclude observations with missing values on any of the variables used in the likelihood calculation. It is up to the user to exclude these observations using if or in.

`nooffset` can be used together with the `xb`, `linpred` or `mu` options to exclude the offset from the prediction. It will only make a difference if the offset option was used in `gllamm`.

`adoonly` causes all `gllamm` to use only ado-code. This option is not necessary if `gllamm` was run with the `adoonly` option.

`from(`*matrix*`)` specifies a matrix of parameters for which the predictions should be made. The column and equation names will be ignored. Without this option, the parameter estimates from the last `gllamm` model will be used.

2.3.3 Syntax for `gllasim`

After estimating a model using `gllamm`, `gllasim` can be used to simulate responses from the model.

```
gllasim varname [if exp] [in range] [ , y u fac linpred mu outcome(#)
    above(#,...,#) nooffset adoonly from(matrix) us(varname) ]
```

`y` the simulated responses are returned in "varname". This option is only necessary if `u`, `fac`, `linpred` or `mu` are also specified.

`u` the simulated latent variables or random effects are returned in "varname"p1, "varname"p2, etc., where the order of the latent variables is the same as in the call to `gllamm` (in the order of the equations in the `eqs()` option). If the `gllamm` model includes equations for the latent variables (`geqs` and/or `bmatrix`), the simulated disturbances are returned.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

`fac` if the `gllamm` model includes equations for the latent variables (`geqs()` and/or `bmatrix()` options in `gllamm`), `fac` causes the simulated latent variables (e.g. factors) to be returned in "varname"p1, "varname"p2, etc. instead of the disturbances, that is, the latent variables on the left-hand side of the structural model.

`linpred` returns the linear predictor including the fixed and simulated random parts in "varname"p. The offset is included (if there is one in the `gllamm` model) unless the `nooffset` option is specified.

`mu` returns the expected value of the response conditional on the simulated values for the latent variables, e.g. a probability if the responses are dichotmous.

`outcome`(#) specifies the outcome for which the predicted probability should be returned (`mu` option) if there is a nominal response and the `expanded()` option has not been used in `gllamm` (with the `expanded()` option, predicted probabilities are returned for all outcomes).

`above`(#,...,#) specifies the events for which the predicted probabilities should be returned (`mu` option) if there are ordinal responses. The probability of a value higher than that specified is returned for each ordinal response. A single number can be given for all ordinal responses.

`nooffset` can be used together with the `linpred` and `mu` options to exclude the offset from the simulated value. It will only make a difference if the `offset()` option was used in `gllamm`.

`fsample` causes `gllasim` to simulate values for the full sample (except observations excluded due to the if and in options), not just the estimation sample.

`adoonly` causes all `gllamm` to use only ado-code. This option is not necessary if `gllamm` was run with the `adoonly` option.

`from`(*matrix*) specifies a matrix of parameters for which the predictions should be made. The column and equation names will be ignored. Without this option, the parameter estimates from the last `gllamm` model will be used.

`us`(*varname*) specifies that, instead of simulating the latent variables, `gllasim` should use the variables in "varname"1, "varname"2, etc.

2.3.4 Some comments on quadrature

Ordinary 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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 (2003) 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' realization 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 number of programs for generalized linear mixed models, including MLwiN and HLM use MQL/PQL (Breslow and Clayton, 1993). Taylor expansions are used to linearize 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, 1996, 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.* (2001b), 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.

Adaptive quadrature

A method that works better than ordinary quadrature is adaptive quadrature (e.g., Liu and Pierce, 1994) implemented in SAS PROC NLMIXED. This method has been implemented in `gllamm`, see Rabe-Hesketh, Skrondal and Pickles (2002; 2005). Rabe-Hesketh, Skrondal and Pickles (2005) show that adaptive quadrature works very well for dichotomous responses with a wide range of cluster sizes and intraclass correlations. Performance of adaptive quadrature is much better than ordinary quadrature, particularly for large cluster sizes and large intraclass correlations. Furthermore, adaptive quadrature usually requires fewer quadrature points than ordinary quadrature to obtain the same precision. For normally distributed responses and counts we would recommend always using adaptive quadrature.

Although adaptive quadrature is likely to give good estimates for continuous responses as long as enough quadrature points are used, it is certainly more computationally efficient to use software that does not use any approximations for this particular case (e.g. Mplus by Muthén and Muthén (1998), S-Plus lme, MLwiN (Goldstein, 1998), etc.)

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

2.3.5 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, Pickles and Skrondal (2003a).

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 3

Multilevel generalized 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 computerized 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 analyze 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 analyzed in Rabe-Hesketh *et al.* (2001b).)

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

$$\eta_{ijk} = \beta_0 + \beta_1 x_{Rijk} + \beta_2 x_{Sijk} + \beta_3 x_{Lijk} + \eta^{(2)}_{jk} + \eta^{(3)}_{k} \tag{3.1}$$

where x_{Rijk} and x_{Sijk} are dummy variable for the relatives and patients with schizophrenia, respectively, x_{Lijk} is the level of difficulty, $\eta^{(2)}_{jk}$ the random intercept for subject j in family k and $\eta^{(3)}_{k}$ is the random intercept for family k. The random intercepts 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` (dichotomized 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.

```
. sort id

. list id famnum group level dtlm in 19/27, clean

        id   famnum   group   level   dtlm
 19.     7        6       3      -1      1
 20.     7        6       3       1      0
 21.     7        6       3       0      0
 22.     8       15       3       0      0
 23.     8       15       3       1      0
 24.     8       15       3      -1      1
```

COBRA
BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

```
 25.    9       10       3       0      0
 26.    9       10       3      -1      0
 27.    9       10       3       1      0
```

3.1.2 Model fitting

A two-level model with a random intercept 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.

In general, we would however recommend using adaptive quadrature by specifying the `adapt` option

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

Here we are using the default of 8 quadrature points because adaptive quadrature tends to require fewer quadrature points. The following output is obtained:

```
. xi: gllamm dtlm i.group level, i(id) fam(binom) link(logit) adapt
i.group           _Igroup_1-3         (naturally coded; _Igroup_1 omitted)
Running adaptive quadrature
Iteration 0:    log likelihood = -310.89641
Iteration 1:    log likelihood = -306.86359
Iteration 2:    log likelihood = -305.96838
Iteration 3:    log likelihood = -305.95923
Iteration 4:    log likelihood = -305.95923
Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -305.95923
Iteration 1:   log likelihood = -305.95923  (backed up)

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

Condition Number = 4.4746866

gllamm model

log likelihood = -305.95923

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Igroup_2 |  -.1691618   .3343253    -0.51   0.613    -.8244273    .4861037
   _Igroup_3 |  -1.023004    .393953    -2.60   0.009    -1.795137   -.2508701
       level |  -1.649218   .1934261    -8.53   0.000    -2.028326    -1.27011
       _cons |   -1.48306   .2836532    -5.23   0.000     -2.03901    -.92711
------------------------------------------------------------------------------


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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
***level 2 (id)

    var(1): 1.6768915 (.66262434)
------------------------------------------------------------------------------
```

The iteration log shows that adaptive quadrature took four iterations to converge. In the iterations under the heading `Adaptive quadrature has converged, running Newton-Raphson` (here just a single iteration), the quadrature locations and weights are held fixed while the parameters are updated by Newton Raphson until Stata's strict convergence criteria are met.

After the iteration log, 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 $\widehat{\beta}_0$ to $\widehat{\beta}_3$ are given in the familiar format used in all of Stata's estimation commands. Group 3 (subjects with schizophrenia) performs considerably worse than group 1 (healthy controls), and performance worsens as difficulty increases as expected. The variance of the level 2 or subject level (id) random effect is estimated as 1.677 with a standard error of 0.663.

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) fam(binom) link(logit) adapt 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
   id1: _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
```

A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
                                                    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 10 Oct 2004 at 17:29:08
Running adaptive quadrature
------------------------------------------------------------------------------
Iteration 0 of adaptive quadrature:
Initial parameters:
         dtlm:      dtlm:      dtlm:      dtlm:       id1:
    _Igroup_2  _Igroup_3      level      _cons      _cons
y1  -.1396641  -.8313329  -1.313382  -1.160498         .5
Updated log likelihood:
-310.89641 -310.89641
                                                   log likelihood = -310.89641
------------------------------------------------------------------------------
Iteration 1 of adaptive quadrature:
Updated parameters:
         dtlm:      dtlm:      dtlm:      dtlm:       id1:
    _Igroup_2  _Igroup_3      level      _cons      _cons
y1  -.1511035  -.9209335  -1.478313  -1.310037   1.246417
Updated log likelihood:
-306.86086 -306.86086 -306.86359 -306.86359
                                                   log likelihood = -306.86359
------------------------------------------------------------------------------

>>>> 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. Between Newton-Raphson steps to update the parameters, `gllamm` goes through a set of iterations to recalculate the adaptive quadrature locations and weights for the new parameter values. This is indicated by the output under `Updated log likelihood` since a change in the quadrature locations and weights results in a small change in the log likelihood. Details of this implementation of adaptive quadrature are given in Rabe-Hesketh, Skrondal and Pickles (2005). We can monitor the change in parameter values and log-likelihood until convergence.

We now consider a model also including a random intercept for families which cannot be estimated in `xtlogit`. This gives a three-level random intercept 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) fam(binom) link(logit)  /*
>   */ nip(5) adapt
i.group           _Igroup_1-3         (naturally coded; _Igroup_1 omitted)
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
Running adaptive quadrature
Iteration 0:    log likelihood = -308.71113
Iteration 1:    log likelihood = -305.36563
Iteration 2:    log likelihood = -305.12406
Iteration 3:    log likelihood =  -305.1227
Iteration 4:    log likelihood = -305.12269

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -305.12269
Iteration 1:   log likelihood = -305.12269

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

Condition Number = 4.2118559

gllamm model

log likelihood = -305.12269

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Igroup_2 |  -.2492026   .3542753    -0.70   0.482    -.9435695    .4451643
   _Igroup_3 |  -1.052524   .3998724    -2.63   0.008     -1.83626   -.2687888
       level |  -1.648225   .1930166    -8.54   0.000     -2.02653   -1.269919
       _cons |  -1.485191   .2845342    -5.22   0.000    -2.042868   -.9275142
------------------------------------------------------------------------------


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

***level 2 (id)

    var(1): 1.1326044 (.67645928)

***level 3 (famnum)

    var(1): .57097887 (.51902956)
------------------------------------------------------------------------------
```

The variance at level 2 is estimated as 1.133 with a standard error of 0.676, and the variance at level 3 is estimated as 0.571 with a standard error of 0.519. 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 5 points per dimension were sufficient, we can use the commands

```
. matrix a=e(b)
. xi: gllamm dtlm i.group level, i(id famnum) fam(binom) link(logit)  /*
>   */ from(a) nip(8) adapt
i.group           _Igroup_1-3         (naturally coded; _Igroup_1 omitted)
Running adaptive quadrature
Iteration 0:    log likelihood = -305.12041
Iteration 1:    log likelihood = -305.12038

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -305.12038
Iteration 1:   log likelihood = -305.12037

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

COBRA
Collection of Biostatistics
Research Archive

```
Condition Number = 4.2143725

gllamm model

log likelihood = -305.12037

------------------------------------------------------------------------------
        dtlm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   _Igroup_2 |  -.2491993   .3543901    -0.70   0.482    -.9437911    .4453925
   _Igroup_3 |  -1.052788   .3999799    -2.63   0.008    -1.836735   -.2688421
       level |  -1.648209    .193194    -8.53   0.000    -2.026863   -1.269556
       _cons |  -1.485365   .2847921    -5.22   0.000    -2.043548   -.9271829
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): 1.137668 (.68586617)

***level 3 (famnum)

    var(1): .56902514 (.52160721)
------------------------------------------------------------------------------
```

Here, the vector of parameter estimates was first stored in the matrix `a` and then passed to `gllamm` as starting values using the `from()` option. We see that using 8 quadrature points (which takes longer since evaluation of the marginal likelihood requires summing 8×8 terms), the parameter estimates do not change considerably giving us some confidence that the estimates are adequate. We would of course report the estimates based on 8 points.

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 centered) year 3 results, `math3`, with a random intercept and a random coefficient of `math3` for schools. The model can be written as

$$\nu_{ij} = \beta_0 + \beta_1 x_{ij} + \eta_{0j} + \eta_{1j} x_{ij} \tag{3.2}$$

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

where i indexes the pupils and j the schools, x_{ij} is the year 3 result and η_{1j} 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, clean
       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`. Note that the variable pupil identifies particular types of pupils with identical covariate values and not individual pupils.

3.2.2 Model fitting

When any of the random effects are not intercepts, we must specify the sets of variables $\mathbf{z}_m^{(l)}$ whose linear combination $\mathbf{z}_m^{(l)\prime}\boldsymbol{\lambda}_m^{(l)}$ multiplies the random effects. This is done by defining equations prior to running `gllamm` using the `eq` command (still available in Stata 8 but undocumented; see help eq_g). 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 in (3.2), the random intercept η_{0j}, multiplies a variable, `cons` with all elements equal to 1, and the random coefficient η_{1j} multiplies `math3`.

We will allow the random slope and intercept to be correlated (the default). We could use the `nocor` option to specify zero correlation. However, the random effects should generally be allowed to be correlated because the model would otherwise not be invariant to translation of x_{ij} (adding a constant to x_{ij}).

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) w(wt) /*
>    */ adapt
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
Running adaptive quadrature
Iteration 0:    log likelihood =  -2796.836
Iteration 1:    log likelihood = -2762.7913
Iteration 2:    log likelihood = -2757.6183
Iteration 3:    log likelihood = -2757.1326
Iteration 4:    log likelihood = -2757.1326

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2757.1326
Iteration 1:   log likelihood = -2757.1326  (backed up)
Iteration 2:   log likelihood = -2757.0942
Iteration 3:   log likelihood = -2757.0803
Iteration 4:   log likelihood = -2757.0803

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

Condition Number = 14.990189

gllamm model

log likelihood = -2757.0803

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6123977   .0428952    14.28   0.000     .5283246    .6964707
       _cons |   30.59295   .3655914    83.68   0.000      29.8764    31.30949
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

  26.964597 (1.346087)

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


***level 2 (school)

    var(1): 4.5788487 (1.3101525)
    cov(2,1): -.36058883 (.12255962) cor(2,1): -.90985323

    var(2): .03430248 (.0176073)
------------------------------------------------------------------------------
```

The within-school or level 1 residual variance is estimated as 26.96 with a standard error of 1.35

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 is the random slope of `math3`. The random intercept variance, `var(1)`, is therefore estimated as 4.58 with a standard error of 1.31 and the random slope variance, `var(2)`, is estimated as 0.034 with a standard error of 0.018. The covariance between the random intercept and slope is estimated as -0.36 with a standard error of 0.12. This corresponds to a correlation of -0.91.

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:

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 have a normal sampling distribution 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, taking the nonstandard situation into account. The program estimates the Cholesky decomposition of the covariance matrix of the random effects.

In order to carry out the likelihood ratio test, will first save the estimates of the current model using

```
estimates store mod2
```

We can 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) eqs(sch_c) w(wt) adapt from(a) skip

Running adaptive quadrature
Iteration 0:    log likelihood = -2768.3507
Iteration 1:    log likelihood = -2767.9009
Iteration 2:    log likelihood = -2767.9009

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2767.9009
Iteration 1:   log likelihood = -2767.9009  (backed up)
Iteration 2:   log likelihood = -2767.8923
Iteration 3:   log likelihood = -2767.8923

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

Condition Number = 14.334774

gllamm model

log likelihood = -2767.8923

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6088066   .0326428    18.65   0.000     .5448278    .6727854
       _cons |   30.60847   .3489154    87.72   0.000     29.92461    31.29233
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

  28.127214 (1.3728903)

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


***level 2 (school)

    var(1): 4.0268447 (1.189892)
------------------------------------------------------------------------------
```

The likelihood ratio test is obtained using Stata's `lrtest` command

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
. lrtest mod2 .
(log-likelihoods of null models cannot be compared)
likelihood-ratio test                                  LR chi2(2)  =     21.62
(Assumption: . nested in mod2)                         Prob > chi2 =    0.0000
```

showing that the effect of `math3` varies significantly between schools. Note that this test is not valid here because the null is on the border of the parameter space. The correct reference distribution is not the 'naive' chi-square with two degrees of freedom χ^2_2 (one degree for the extra variance and one for the covariance) but rather the mixture $\frac{1}{2}\chi^2_0 + \frac{1}{2}\chi^2_2$. A correct p-value is obtained by halving the p-value from the χ^2_2 distribution.

To make predictions for the previous random coefficient model, we must first 'restore' the estimates (which is faster than re-estimating the model!):

```
estimates restore mod2
```

We can then use `gllapred` with the `u` option to obtain the posterior means (empirical Bayes predictions) of the random effects.

```
. gllapred eb, u
(means and standard deviations will be stored in ebm1 ebs1 ebm2 ebs2)
Non-adaptive log-likelihood: -2757.1446
-2757.5262 -2757.0856 -2757.0803 -2757.0803
log-likelihood:-2757.0803
```

This creates four variables, `ebm1` and `ebm2` containing the posterior means of the random intercept and coefficient, respectively, and `ebs1` and `ebs2` 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  ebm1 ebm2 ebs1 ebs2 in 87/95, clean
       school   pupil         ebm1         ebm2        ebs1        ebs2
 87.        5      21   -.03652739   -.00149369   1.0443018   .11192404
 88.        5      22   -.03652739   -.00149369   1.0443018   .11192404
 89.        5      23   -.03652739   -.00149369   1.0443018   .11192404
 90.        5      24   -.03652739   -.00149369   1.0443018   .11192404
 91.        5      25   -.03652739   -.00149369   1.0443018   .11192404
 92.        6       1    .85082355   -.07551752   1.1132722   .10827583
 93.        6       2    .85082355   -.07551752   1.1132722   .10827583
 94.        6       3    .85082355   -.07551752   1.1132722   .10827583
 95.        6       4    .85082355   -.07551752   1.1132722   .10827583
```

We can summarize the values of `ebm1` and `ebm2` 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 ebm1 ebm2 if f==1
    Variable |       Obs        Mean    Std. Dev.        Min        Max
-------------+----------------------------------------------------------
        ebm1 |        48    2.59e-07    1.853219  -3.834247   3.270769
        ebm2 |        48   -2.08e-08    .1517099  -.2679346    .336618
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
. corr ebm1 ebm2 if f==1, cov
(obs=48)

             |     ebm1      ebm2
-------------+------------------
        ebm1 |  3.43442
        ebm2 | -.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

$$\begin{aligned} \widehat{y}_{ij} &= \widehat{\nu}_{ij} \\ &= \widehat{\beta}_0 + \widehat{\beta}_1 x_{ij} + \widetilde{\eta}_{0j} + \widetilde{\eta}_{1j} x_{ij}, \end{aligned} \tag{3.3}$$

where $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are the fixed parameter estimates and $\widetilde{\eta}_{0j}$ and $\widetilde{\eta}_{1j}$ are the posterior means of the random intercept and slope, respectively.

```
. gllapred lp, linpred
(linear predictor will be stored in lp)
Non-adaptive log-likelihood: -2757.1446
-2757.5262 -2757.0856 -2757.0803 -2757.0803
log-likelihood:-2757.0803
```

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

```
. sort school math3
. twoway (line lp math3, connect(ascending)), ytitle(Predicted math5 score) ///
> xtitle(Math3 score)
```

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 as level 2 residuals. Outlying schools could be detected by dividing the level 2 residuals by the sampling standard deviation (the *diagnostic standard error*). For linear mixed models the squared diagnostic standard error is the random effect (prior) variance minus the posterior variance.

For more examples of estimating linear mixed models using `gllamm`, see the textbook examples for Hox (2002), Kreft and de Leeuw (1998) and Singer and Willett (2003) at `http://www.gllamm.org/examples.html` Section 5.2 discusses a discrete random effects version of the model considered in this section.

Rabe-Hesketh and Everitt (2004) describe how a random coefficient model for dichotomous responses can be estimated using `gllamm`. For an application of a random coefficient model to meta-analysis, see the worked example for Setion 9.5 of Skrondal and Rabe-Hesketh (2004) at http://www.gllamm.org/books/examples.html#genlat

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

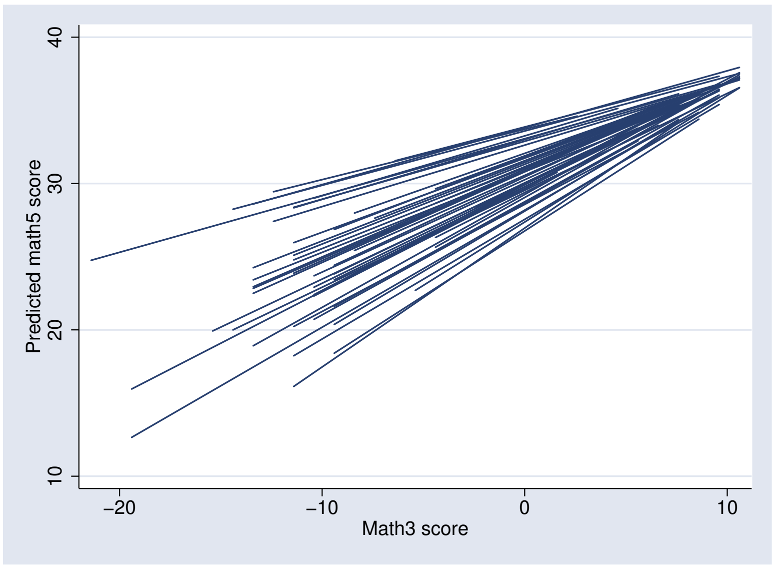


Figure 3.1: Predictions from random coefficients model for JSP data.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 4

Multilevel factor and item response models

4.1 One parameter and two parameter item-response models

Binary data on five 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. We will here consider the one and two-parameter logistic item response models. In the one-parameter logistic model the log odds of subject j giving a correct answer to item i is

$$\nu_{ij} = \beta_i + \eta_j \tag{4.1}$$

where $-\beta_i$ represents the difficulty of question or item i and the 'factor' η_j represents the ability of subject j. This is a simple two-level model and may be fitted using `xtlogit`. Note that the model is known as the Rasch model if the η_j are construed as parameters instead of random variables.

If we introduce a further parameter λ_i, we obtain a two-parameter logistic item response model

$$\nu_{ij} = \beta_i + \eta_j \lambda_i \tag{4.2}$$

where λ_i represents the extent to which item i discriminates between subjects of different abilities. Here we are modeling a multivariate dataset by representing the variables as level 1 units indexed i. 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

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

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

$$\nu_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \eta_j \mathbf{x}'_{ij}\boldsymbol{\lambda}. \tag{4.4}$$

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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 dummies 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, clean
      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 model (see equation (4.1)) may be estimated using

```
. gllamm resp i1 i2 i3 i4 i5, nocons link(logit) fam(bin) i(id) w(wt)/*
>   */ adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -2474.5358
Iteration 1:    log likelihood = -2467.6509
Iteration 2:    log likelihood = -2466.9386
Iteration 3:    log likelihood = -2466.9381

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2466.9381
Iteration 1:   log likelihood = -2466.9378
Iteration 2:   log likelihood = -2466.9376

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

Condition Number = 2.363268

gllamm model

log likelihood = -2466.9376

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          i1 |   2.730012   .1304412    20.93   0.000     2.474352    2.985672
          i2 |   .9986047   .0791771    12.61   0.000     .8434204    1.153789
          i3 |   .2398532   .0717746     3.34   0.001     .0991776    .3805287
          i4 |    1.30645   .0846379    15.44   0.000     1.140563    1.472337
          i5 |   2.099403   .1054449    19.91   0.000     1.892734    2.306071
------------------------------------------------------------------------------


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


***level 2 (id)
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
    var(1): .57022544 (.10486337)
------------------------------------------------------------------------------
```

The factor 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)` if the data is not in 'collapsed' form. However, `xtlogit` uses ordinary quadrature, not adaptive quadrature.)

In order to fit the 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).

```
. local ll = e(ll)
. eq id: i1 i2 i3 i4 i5
. gllamm resp i1 i2 i3 i4 i5, nocons link(logit) fam(bin) i(id)/*
>  */ eqs(id) w(wt) lf0(6 `ll') adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -2473.0647
Iteration 1:    log likelihood = -2468.8954
Iteration 2:    log likelihood = -2466.8525
Iteration 3:    log likelihood = -2466.6613
Iteration 4:    log likelihood = -2466.6613

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2466.6613
Iteration 1:   log likelihood = -2466.6613  (backed up)
Iteration 2:   log likelihood = -2466.6534
Iteration 3:   log likelihood = -2466.6533

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

Condition Number = 11.498716


gllamm model                                      Number of obs   =       5000
                                                  LR chi2(4)      =       0.57
Log likelihood = -2466.6533                       Prob > chi2     =     0.9665

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          i1 |   2.773246   .2057431    13.48   0.000     2.369997    3.176495
          i2 |   .9901996   .0900181    11.00   0.000     .8137673    1.166632
          i3 |     .24915   .0762745     3.27   0.001     .0996548    .3986452
          i4 |   1.284755   .0990362    12.97   0.000     1.090647    1.478862
          i5 |   2.053265   .1353571    15.17   0.000      1.78797     2.31856
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): .68174302 (.42622871)

    loadings for random effect 1
    i1: 1 (fixed)
    i2: .87532845 (.36270233)
    i3: 1.0790061 (.43516089)
    i4: .83369313 (.36726473)
    i5: .79552018 (.38060867)

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

Collection of Biostatistics Research Archive

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.

We can obtain empirical Bayes predictions of the latent trait (ability) using

```
. gllapred score, u
(means and standard deviations will be stored in scorem1 scores1)
Non-adaptive log-likelihood: -2466.6532
-2466.6533 -2466.6533
log-likelihood:-2466.6533
```

The variable `scorem1` contains the posterior mean (the empirical Bayes prediction) and `scores1` the corresponding posterior standard deviation.

In the above model, the factor loading of item 1 was constrained to 1 and the variance of the latent variable 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(.68174302)`=.82567731 as the first loading and multiply all other loadings by this value.

We can also estimate the model using this alternative parameterization way by using the `frload()` option to free the first factor loading and the `contraints` option to set the variance of the latent variable to 1. In order to use `constraints`, we have to know the equation and column name for the parameter of interest. One way to find this is to simply display the matrix of parameter estimates:

```
. matrix list e(b)

e(b)[1,10]
         resp:      resp:      resp:      resp:      resp:    id1_1l:    id1_1l:    id1_1l:    id1_1l:
           i1         i2         i3         i4         i5         i2         i3         i4         i5
y1  2.7732456  .99019959  .24914998  1.2847549  2.0532647  .87532845  1.0790061  .83369313  .79552018

        id1_1:
           i1
y1  .82567731
```

The last parameter is the standard deviation of the latent variable. We can constrain this to 1 by defining the constraint

```
constraint def 1 [id1_1]i1 = 1
```

and passing it to the `gllamm` command as follows:

```
. gllamm resp i1 i2 i3 i4 i5, nocons link(logit) fam(bin) i(id) eqs(id) w(wt) /*
> constr(1) frload(1) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -2492.9919
Iteration 1:    log likelihood = -2473.2543
Iteration 2:    log likelihood = -2467.2626
Iteration 3:    log likelihood = -2466.6723
Iteration 4:    log likelihood = -2466.6547
Iteration 5:    log likelihood = -2466.6547

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2466.6547
Iteration 1:   log likelihood = -2466.6547  (backed up)
Iteration 2:   log likelihood = -2466.6533
Iteration 3:   log likelihood = -2466.6533
```

Collection of Biostatistics
Research Archive

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

Condition Number = 4.6103103

gllamm model with constraints:
 ( 1)  [id1_1]i1 = 1

log likelihood = -2466.653343760208

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          i1 |   2.773234    .205743    13.48   0.000     2.369985    3.176483
          i2 |   .9901996   .0900182    11.00   0.000     .8137672    1.166632
          i3 |     .24915   .0762746     3.27   0.001     .0996546    .3986454
          i4 |   1.284755   .0990363    12.97   0.000     1.090647    1.478862
          i5 |   2.053265   .1353574    15.17   0.000      1.78797    2.318561
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): 1 (0)

    loadings for random effect 1
    i1: .82565942 (.25811315)
    i2: .72273928 (.18667773)
    i3: .890914 (.2328178)
    i4: .68836241 (.18513868)
    i5: .65684452 (.20990788)

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

If we obtain empirical Bayes predictions from this model, they will be perfectly correlated (but scaled differently):

```
. gllapred nscore, u
(means and standard deviations will be stored in nscorem1 nscores1)
Non-adaptive log-likelihood: -2466.6532
-2466.6533 -2466.6533
log-likelihood:-2466.6533
. corr scorem1 nscorem1
(obs=160)
             |  scorem1 nscorem1
-------------+------------------
     scorem1 |   1.0000
    nscorem1 |   1.0000   1.0000
```

Section 8.2 shows how item-response models for ordinal data can be estimated in `gllamm`. See also the worked example for Section 9.4 of Skrondal and Rabe-Hesketh (2004) at http://www.gllamm.org/examples.html#genlat

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 5

Discrete random effects

5.1 A simple finite mixture model

In the Handbook of Statistical Analyses using Stata, Rabe-Hesketh and Everitt (2004) describe finite mixture modeling 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,

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

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 η_i, y_i has a normal distribution with variance σ^2 and expectation

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

where

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

and η_i is a discrete latent variable with two values, $e_1 = \mu_1 - \mu$ and $e_2 = \mu_2 - \mu$ where μ is the overall mean.

5.1.1 Data preparation

First we read the data

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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

```
gen id=_n
```

The first ten ages are:

```
. list id ages in 1/10, clean

       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)

number of level 1 units = 99

Condition Number = 16.449916

gllamm model

log likelihood = -383.39585

------------------------------------------------------------------------------
        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
------------------------------------------------------------------------------

  135.29986 (19.230685)
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

The mean is estimated as 30.47, the variance as 135.30 and the log-likelihood is −383.40.

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 maximize 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)
..............................................................................
> ......................
maximum gateaux derivative is .04365856
Iteration 0:   log likelihood = -382.93311  (not concave)
Iteration 1:   log likelihood = -382.74883  (not concave)
Iteration 2:   log likelihood = -380.90691
Iteration 3:   log likelihood = -375.43929
Iteration 4:   log likelihood = -375.27624
Iteration 5:   log likelihood = -373.71499
Iteration 6:   log likelihood =  -373.6975
Iteration 7:   log likelihood = -373.69749

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

Condition Number = 23.174022


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
------------------------------------------------------------------------------

  44.646396 (7.853091)

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


***level 2 (id)

    loc1: 16.426, -5.5187
  var(1): 90.653482
    prob: 0.2515, 0.7485
------------------------------------------------------------------------------
```

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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 (2004), hardly increases the log-likelihood, so for these data the assumption of equal variances is appropriate.)

To see how the model is parameterized and obtain standard errors for all parameters, 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   .0879476    21.60   0.000     1.727013    2.071761
-------------+----------------------------------------------------------------
z2_1_1       |
       _cons |   16.42646   1.852971     8.86   0.000      12.7947    20.05822
-------------+----------------------------------------------------------------
p2_1         |
       _cons |  -1.090743   .2742902    -3.98   0.000    -1.628342   -.5531437
------------------------------------------------------------------------------
```

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

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

and z2_1_1 is the location for class 1

$$\widehat{e}_1 = \mu_1 - \mu = \texttt{z2\_1\_1}. \tag{5.6}$$

The location for class 2 is estimated as

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

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

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

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

```
. gllapred prob, p
(probabilities will be stored in prob1 prob2)
prior probabilities
   prob: 0.2515, 0.7485

locations for random effect 1
   loc: 16.43, -5.519

log-likelihood:-373.69749
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

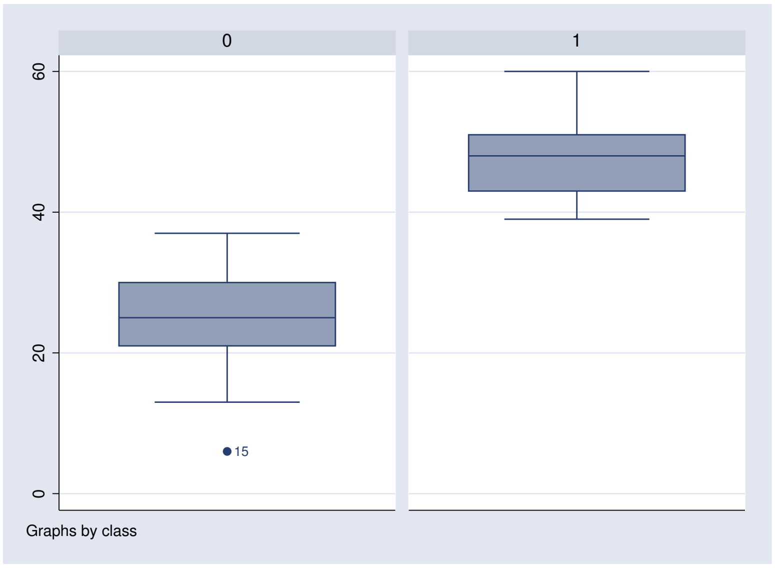


Figure 5.1: Boxplots of ages of onset by assigned latent class

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 `prob1` and `prob2`.

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

```
gen class = prob2>prob1
sort class
graph box ages, medtype(line) by(class) ///
marker(1, mlabel(id)) marker(1, mlabel(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) s(het)
Iteration 0:   log likelihood = -411821.02  (not concave)
Iteration 1:   log likelihood = -400.19611
Iteration 2:   log likelihood = -389.57669
Iteration 3:   log likelihood = -383.41147
Iteration 4:   log likelihood = -383.39585
Iteration 5:   log likelihood = -383.39585

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

Condition Number = 16.449916

gllamm model

log likelihood = -383.39585
```

COBRA
Collection of Biostatistics Research Archive
Hosted by The Berkeley Electronic Press

```
No fixed effects

Variance at level 1
------------------------------------------------------------------------------

  135.29988 (19.230687)

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


***level 2 (id)

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

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) s(het) lf0(`k' `ll') /*
>    */ gateaux(20 60 100) from(a)
...............................................................................
> .....................
maximum gateaux derivative is .04286811
Iteration 0:   log likelihood = -383.27324
Iteration 1:   log likelihood = -382.49403  (not concave)
Iteration 2:   log likelihood = -379.58411
Iteration 3:   log likelihood = -375.05356
Iteration 4:   log likelihood = -373.70929
Iteration 5:   log likelihood =  -373.6975
Iteration 6:   log likelihood = -373.69749

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

Condition Number = 20.454284

gllamm model

log likelihood = -373.69749

No fixed effects

Variance at level 1
------------------------------------------------------------------------------

  44.646404 (7.8530935)

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


***level 2 (id)

    loc1: 46.901, 24.956
  var(1): 90.653454
    prob: 0.2515, 0.7485
------------------------------------------------------------------------------
```

A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

5.2 Linear mixed model with discrete random effects

In this section we will return to the Junior School Project data analyzed 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 regressing the year 5 results, `math5`, on the (mean centered) year 3 results, `math3`, with a random intercept and a random coefficient of `math3` for schools. The model can be written as

$$\nu_{ij} = \beta_0 + \beta_1 x_{ij} + \eta_{0j} + \eta_{1j} x_{ij} \tag{5.9}$$

where i indexes the pupils and j the schools, x_{ij} is the year 3 result and η_{1j} is the corresponding random coefficient. Instead of assuming a bivariate normal distribution of the random effects as in Section 3.2, we now assume a bivariate discrete distribution, i.e., we assume that the random effects (η_{0j}, η_{1j}) take on a number of discrete values (e_{0r}, e_{1r}), with probabilities π_r, $r = 1, \ldots, R$. This corresponds to assuming that the population falls into a finite number of latent classes or types or can be approximated in this way. The model is also known as a *mixture regression model.* When the maximum number of classes is used (so that the likelihood cannot be increased by adding more classes), the estimates may be interpreted as non-parametric maximum likelihood estimates.

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)
Iteration 0:   log likelihood = -2972.4441  (not concave)
Iteration 1:   log likelihood = -2844.1229  (not concave)
Iteration 2:   log likelihood = -2800.2744  (not concave)
Iteration 3:   log likelihood = -2767.3842
Iteration 4:   log likelihood = -2766.6151  (not concave)
Iteration 5:   log likelihood = -2761.2258
Iteration 6:   log likelihood = -2760.7652
Iteration 7:   log likelihood = -2760.7036
Iteration 8:   log likelihood = -2760.7033

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

Condition Number = 25.665755

gllamm model

log likelihood = -2760.7033

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .5921935   .0397523    14.90   0.000      .5142805    .6701065
       _cons |   30.71605   .3245171    94.65   0.000      30.08001    31.35209
------------------------------------------------------------------------------
```

Collection of Biostatistics Research Archive

```
Variance at level 1
------------------------------------------------------------------------------

  28.171487 (1.3495007)

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


***level 2 (school)

    loc1: -1.2586, 2.5874
  var(1): 3.2566258

    loc2: .11034, -.22684
cov(2,1): -.2855053
  var(2): .02502998
    prob: 0.6727, 0.3273
------------------------------------------------------------------------------
```

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 maximized 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 if 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(3) weight(wt) /*
>  */ ip(f) from(a) gateaux(-10 10 30) lf0(`k' `ll')
.................................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> .....................................................
maximum gateaux derivative is .85177794
Iteration 0:   log likelihood =  -2758.317  (not concave)
Iteration 1:   log likelihood = -2758.2569
Iteration 2:   log likelihood =  -2757.502
Iteration 3:   log likelihood = -2757.4439
Iteration 4:   log likelihood = -2757.4437

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

Condition Number = 60.948278
```

COBRA
Collection of Biostatistics
Research Archive

```
gllamm model                                      Number of obs   =        887
                                                  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)

    loc1: -1.1642, -2.6896, 2.6625
  var(1): 3.33719

    loc2: .07502, .91947, -.24032
cov(2,1): -.33125196
  var(2): .04652579
    prob: 0.6551, 0.0291, 0.3158
------------------------------------------------------------------------------
```

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 (-2.6896,0.91947) 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) w(wt) ip(f) /*
>   */ from(a) gateaux(-10 10 30) lf0(`k' `ll')
.................................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...............................................................................
> ...................................................
maximum gateaux derivative is .11892368

Iteration 0:   log likelihood =  -2756.919  (not concave)
Iteration 1:   log likelihood = -2756.8514  (not concave)
Iteration 2:   log likelihood = -2753.7822
Iteration 3:   log likelihood = -2752.2336
Iteration 4:   log likelihood = -2751.4414
Iteration 5:   log likelihood = -2751.1913
Iteration 6:   log likelihood = -2751.1612
Iteration 7:   log likelihood = -2751.1611
```

Collection of Biostatistics
Research Archive

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

Condition Number = 55.416924


gllamm model                                     Number of obs   =        887
                                                 LR chi2(3)      =      12.57
Log likelihood = -2751.1611                      Prob > chi2     =     0.0057

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6169153   .0457245    13.49   0.000     .5272968    .7065337
       _cons |   30.65181   .3611107    84.88   0.000     29.94405    31.35958
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

  26.644568 (1.2922175)

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


***level 2 (school)

    loc1: -.34235, -2.6291, -3.3806, 2.9257
  var(1): 4.4624483

    loc2: .06313, .88986, .08057, -.27122
cov(2,1): -.34745305
  var(2): .04844958
    prob: 0.5334, 0.0316, 0.1597, 0.2753
------------------------------------------------------------------------------
```

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. Before doing this, we save the current estimates using

```
estimates store mod1
```

We can now re-estimate the three point model as follows:

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

a[1,12]
          math5:      math5:       lns1:      z2_1_1:     z2_2_1:        p2_1:     z2_1_2:
          math3       _cons       _cons        cons       math3       _cons        cons
y1    .61691526   30.651812   1.6412926   -.3423471    .0631317   .66136261  -2.6290895

         z2_2_2:       p2_2:      z2_1_3:     z2_2_3:        p2_3:
          math3       _cons        cons       math3       _cons
y1    .88986167  -2.1642929  -3.3805714    .08057202  -.54481331

. 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 /*
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
>  */ weight(wt) ip(f)
Iteration 0:   log likelihood = -2755.6625  (not concave)
Iteration 1:   log likelihood =    -2753.5
Iteration 2:   log likelihood = -2753.4706
Iteration 3:   log likelihood = -2753.4705

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

Condition Number = 37.366097

gllamm model

log likelihood = -2753.4705

------------------------------------------------------------------------------
       math5 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       math3 |   .6071674    .040136    15.13   0.000     .5285023    .6858326
       _cons |    30.6246   .3660524    83.66   0.000     29.90715    31.34205
------------------------------------------------------------------------------

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

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


***level 2 (school)

    loc1: -.32603, -3.4409, 2.9461
  var(1): 4.6435916

    loc2: .07632, .16678, -.26104
cov(2,1): -.33181452
  var(2): .02708821
    prob: 0.539, 0.1851, 0.2759
------------------------------------------------------------------------------
```

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

We now restore the estimates of the four class model using

```
estimates restore mod1
```

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
(probabilities will be stored in z1 z2 z3 z4)
prior probabilities
    prob: 0.5334, 0.0316, 0.1597, 0.2753

locations for random effect 1
     loc: -.3423, -2.629, -3.381, 2.926

locations for random effect 2
     loc: .0631, .8899, .0806, -.2712

log-likelihood:-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`.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 posterior probability using

```
. egen double maxp = rmax(z1 z2 z3 z4)
. summ maxp
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        maxp |       848    .8940588    .1446455   .5110531   .9998172
```

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
(means and standard deviations will be stored in zm1 zs1 zm2 zs2)
```

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, clean
       school   pupil          zm1          zm2         zs1         zs2
 87.        5      21   -.28264137    .05583627   .52128051   .04961075
 88.        5      22   -.28264137    .05583627   .52128051   .04961075
 89.        5      23   -.28264137    .05583627   .52128051   .04961075
 90.        5      24   -.28264137    .05583627   .52128051   .04961075
 91.        5      25   -.28264137    .05583627   .52128051   .04961075
 92.        6       1    .59826102   -.03395675   1.4983686   .15183621
 93.        6       2    .59826102   -.03395675   1.4983686   .15183621
 94.        6       3    .59826102   -.03395675   1.4983686   .15183621
 95.        6       4    .59826102   -.03395675   1.4983686   .15183621
```

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:

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
. 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 |  -.2103802   .02200343          28
        2 | -3.6379418   .33661802           1
        3 |  -2.707364   .20038459           7
        4 |  2.3733456  -.19628394          12
----------------------------------------------
```

The covariance matrices of the two sets of posterior means are

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

             |      zm1      zm2
-------------+------------------
         zm1 |  3.56472
         zm2 | -.280941  .037932
. corr um1 um2 if f==1, cov
(obs=48)

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

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 were created using the commands

```
twoway (scatter um1 zm1 if f==1), ytitle(Quadrature) xtitle(Free masses)

twoway (scatter um2 zm2 if f==1), ytitle(Quadrature) xtitle(Free masses)
```

An example of an exploratory latent class model for rankings is given in Section 9.4. Further examples of latent class models using `gllamm` are available at `http://fmwww.bc.edu/repec/nasug2003/lclass.pdf`.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

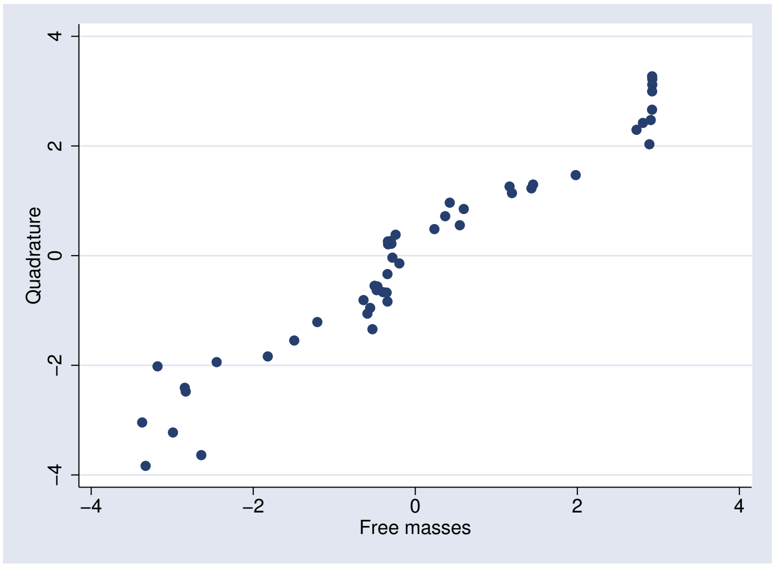


Figure 5.2: Posterior means of random intercept: quadrature solution versus discrete solution

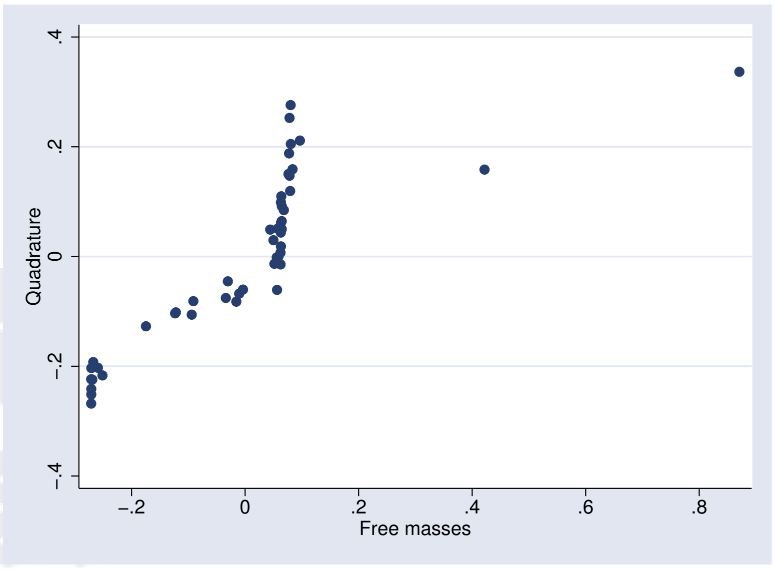


Figure 5.3: Posterior means of random slope: quadrature solution versus discrete solution

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 6

Mixed response models

The models we have discussed so far have had responses of a single type, dichotomous 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 fiber intake (exposure) and risk of CHD (disease) where fiber is subject to measurement error and has been measured twice on a subset of subjects.

We therefore have several responses y_{ij} per subject j, one or two of the fallible measure of exposure y_{1j}, y_{2j} and disease status y_{3j}. 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_{1j} and y_{2j} are independently normally distributed conditional on true exposure with means μ_{ij} given by

$$\begin{aligned} \nu_{ij} = \mu_{ij} &= \beta_i + \eta_j \lambda_i, \quad i = 1, 2 \\ &= \beta_1 + \eta_j \end{aligned} \tag{6.1}$$

where the mean exposure on both occasions is assumed to be the same ($\beta_1 = \beta_2$), η_j is a latent variable representing the difference between subject jth'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 constraining the factor loadings to 1, we ensure that the scale of η_j is the same as that of the exposure measurements.

We now specify a *disease model* by assuming that y_{3j} is binomial with the logit of the probability π_{3j} given by

$$\nu_{3j} = \text{logit}(\pi_{3j}) = \beta_3 + \eta_j \lambda_3. \tag{6.2}$$

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.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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

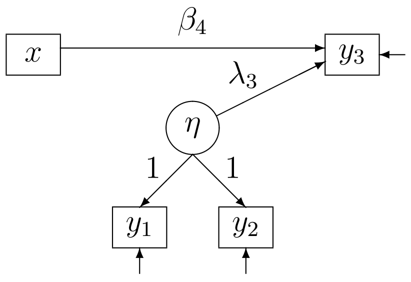
$$\begin{aligned}\nu_{ij} &= \beta_1 z_{1i} + \beta_3 z_{3i} + \eta_j(z_{1i} + \lambda_3 z_{3i}) \\ &= \beta_i + \eta_j \lambda_i \end{aligned} \tag{6.3}$$

where $z_{1i} = 1$ if $i = 1$ or $i = 2$ and 0 otherwise and $z_{3i} = 1$ if $i = 3$ 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_{1i} 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_j, to the disease model

$$\nu_{3j} = \text{logit}(\pi_{3j}) = \beta_3 + \beta_4 x_j + \eta_j \lambda_3 \tag{6.4}$$

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 variability.)

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

$$\eta_j = \gamma x_j + \zeta_j, \tag{6.5}$$

where ζ_j is a residual error term or disturbance. The measurement model now is

$$\nu_{ij} = \beta_i + \gamma x_j + \zeta_j, \tag{6.6}$$

and the disease model is

$$\nu_{3j} = \beta_3 + \gamma \lambda_3 x_j + \zeta_j \lambda_3. \tag{6.7}$$

The coefficient of x_j (representing the indirect effect of x_j on the risk of disease), $\gamma\lambda_3$, in the disease model is the product of the coefficient of x_j 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 η_j on x_j in (6.5) directly. The model is shown as a path diagram below:

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

$$\begin{tikzcd}
\boxed{x} \arrow[drr, "\gamma"] & & & & \boxed{y_3} & {} \arrow[l] \\
& & \eta \arrow[urr, "\lambda_3"'] \arrow[dl, "1"'] \arrow[dr, "1"] & {} \arrow[l] & & \\
& \boxed{y_1} & & \boxed{y_2} & & \\
& {} \arrow[u] & & {} \arrow[u] & &
\end{tikzcd}$$

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

$$\nu_{3j} = \beta_3 + (\beta_4 + \gamma\lambda_3)x_j + \zeta_j\lambda_3. \tag{6.8}$$

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 explicitly 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:

$$\begin{tikzcd}
\boxed{x} \arrow[rrrr, "\beta_4"] \arrow[drr, "\gamma"] & & & & \boxed{y_3} & {} \arrow[l] \\
& & \eta \arrow[urr, "\lambda_3"'] \arrow[dl, "1"'] \arrow[dr, "1"] & {} \arrow[l] & & \\
& \boxed{y_1} & & \boxed{y_2} & & \\
& {} \arrow[u] & & {} \arrow[u] & &
\end{tikzcd}$$

6.1.1 Data preparation

The data are in *diet.dta*. The variable `r` contains the logarithm of the dietary fiber measurements and `chd` is the binary disease indicator. Those subjects who had two fiber measurements have two lines of data; the variable `t` indicates whether `r` corresponds to the first measurement of fiber (`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, clean

        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
```

COBRA
BERKELEY ELECTRONIC REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

Note that, for instance, subject 214 had only one measurement of fiber 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 fiber 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 fiber and the variables `diet` and `chd` indicate whether the observation in `resp` is log fiber 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 clean
        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.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

6.1.2 Parameter estimation

We now specify a factor model where η_j is the factor with mixed responses, continuous fiber intake and dichotomous CHD status. We must specify an equation to define the variables whose linear combination ($\mathbf{z}'\boldsymbol{\lambda}$ 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 fiber 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) /*
>         */ adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -318.10383
Iteration 1:    log likelihood = -207.50866
Iteration 2:    log likelihood = -196.60196
Iteration 3:    log likelihood = -194.15238
Iteration 4:    log likelihood = -187.15595
Iteration 5:    log likelihood = -186.93098
Iteration 6:    log likelihood = -186.93042
Iteration 7:    log likelihood = -186.93042

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -186.93042
Iteration 1:   log likelihood = -186.93042

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

Condition Number = 56.157964

gllamm model

log likelihood = -186.93042

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |   2.863709   .0238868   119.89   0.000     2.816892    2.910526
         chd |  -1.977035   .2571137    -7.69   0.000    -2.480968   -1.473101
        occc |   .0471733   .3328395     0.14   0.887    -.6051802    .6995268
        occd |  -.1208468   .0327979    -3.68   0.000    -.1851296   -.0565641
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------
  .02173263 (.00352693)
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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


***level 2 (id)

    var(1): .07019984 (.00758013)

    loadings for random effect 1
    diet: 1 (fixed)
    chd: -1.9570825 (.72619423)

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

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

```
. disp exp(-1.9570825)
.14126998
```

The effect of occupation on diet is estimated as -0.12 with bus staff eating less fiber 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 occupation 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 unidimensional 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:      id1_1l:
                  diet          chd         occc         occd        _cons          chd
 resp:diet           1
  resp:chd   -.1481915            1
 resp:occc    .1147171   -.71970615            1
 resp:occd  -.72830275    .10791228   -.15269819            1
lns1:_cons  -.02022821   -.01695725     .0023973     .0147321            1
id1_1l:chd   .00302364    .26864422   -.01220304   -.00226067   -.11648136            1
id1_1:diet   .01722283    .00907847    -.0029056   -.01253225   -.40845601    .09658715

                id1_1:
                  diet
id1_1:diet           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 fiber measurements are multiplied by 2. The parameter estimates change as expected (e.g. the log odds and its standard error halve), confirming that the parameter estimates can be trusted.

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

```
eq f1: occ
```

A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 must omit occupation from the measurement model:

```
. gllamm resp diet chd occc, /*
>         */ nocons i(id) eqs(id) link(ident logit) /*
>         */ family(gauss binom) lv(var) fv(var) /*
>         */ adapt geqs(f1)
Running adaptive quadrature
Iteration 0:    log likelihood = -321.69211
Iteration 1:    log likelihood = -208.17433
Iteration 2:    log likelihood = -193.47061
Iteration 3:    log likelihood = -187.63514
Iteration 4:    log likelihood = -186.93259
Iteration 5:    log likelihood = -186.93042
Iteration 6:    log likelihood = -186.93042

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -186.93042
Iteration 1:   log likelihood = -186.93042  (backed up)

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

Condition Number = 56.526515

gllamm model

log likelihood = -186.93042

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |   2.863724   .0238865   119.89   0.000     2.816907    2.910541
         chd |  -1.977149   .2571253    -7.69   0.000    -2.481105   -1.473193
        occc |  -.1893826   .3396352    -0.56   0.577    -.8550554    .4762901
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

  .02173221 (.00352686)

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


***level 2 (id)

    var(1): .07019764 (.00757977)

    loadings for random effect 1
    diet: 1 (fixed)
    chd: -1.9568884 (.72622195)


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

    random effect 1 has 1 covariates:
    occ: -.1208696 (.03279744)
------------------------------------------------------------------------------
```

The estimate of the direct effect of occupation on diet $\widehat{\beta}_4$ is negative though not significant. For the same fiber intake, bus staff are at reduced risk of heart disease; combined with the

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

increased risk due to lower fiber intake, the total effect of occupation on diet ($\beta_4 + \gamma\lambda_3$) is negligible as we saw using the previous parametrization.

Direct effect of diet on heart disease

By omitting the `geqs()` option, we can estimate the model with no effect of occupation on fiber 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) /*
>          */ adapt from(a) skip
Running adaptive quadrature
Iteration 0:    log likelihood =   -201.424
Iteration 1:    log likelihood = -193.77342
Iteration 2:    log likelihood = -193.59706
Iteration 3:    log likelihood = -193.59671
Iteration 4:    log likelihood =  -193.5967

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood =  -193.5967
Iteration 1:   log likelihood =  -193.5967

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

Condition Number = 54.790434

gllamm model

log likelihood = -193.5967

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.       z     P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |   2.799603    .016685    167.79   0.000     2.766901    2.832305
         chd |  -1.874168   .2480817     -7.55   0.000    -2.360399   -1.387937
        occc |  -.1408247   .3353358     -0.42   0.675    -.7980707    .5164213
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

  .02188091 (.00357804)

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


***level 2 (id)

    var(1): .07356131 (.00787928)

    loadings for random effect 1
    diet: 1 (fixed)
    chd: -1.9413481 (.72028026)

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

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 fiber intake.

```
. gllamm resp diet chd, /*
>         */ nocons i(id) eqs(id) link(ident logit) /*
>         */ family(gauss binom) lv(var) fv(var) /*
>         */ adapt from(a) geqs(f1) skip
Running adaptive quadrature
Iteration 0:    log likelihood = -187.31544
Iteration 1:    log likelihood =  -187.0855
Iteration 2:    log likelihood = -187.08527
Iteration 3:    log likelihood = -187.08525

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -187.08525
Iteration 1:   log likelihood = -187.08524
Iteration 2:   log likelihood = -187.08523

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

Condition Number = 54.728084

gllamm model

log likelihood = -187.08523

------------------------------------------------------------------------------
        resp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        diet |     2.8634   .0238872   119.87   0.000     2.816582    2.910218
         chd |  -2.068703   .2015013   -10.27   0.000    -2.463638   -1.673768
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

  .02168813 (.00351046)

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


***level 2 (id)

    var(1): .07026323 (.00757039)

    loadings for random effect 1
    diet: 1 (fixed)
    chd: -1.8619302 (.70226499)


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

    random effect 1 has 1 covariates:
    occ: -.12017981 (.03278624)
------------------------------------------------------------------------------
```

Rabe-Hesketh, Pickles and Skrondal (2003a) estimated a semi-parametric mixture model for this dataset using `gllamm`. The approach is also often referred to as nonparametric maximum likelihood estimation. Rabe-Hesketh, Skrondal and Pickles (2003b) discuss estimation of these models using `gllamm` and introduce a `gllamm` wrapper called `cme`. The wrapper for generalized linear models with covariate measurement error accepts a simple syntax, performs

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

all the required data manipulation, calls `gllamm` and then produces user-friendly output; see `http://www.gllamm.org/wrappers.html`. The wrapper can also be used to print out all the commands required for the data manipulation and for estimating the model in `gllamm`.

Worked examples for several other mixed response models from Chapter 14 of Skrondal and Rabe-Hesketh (2004) are available at `http://www.gllamm.org/examples.html#genlat`

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 modeled as

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

where t is time, $h^0(t)$ is the 'baseline' hazard and ν_{ij} is the linear predictor of GLLAMMs. Here we have used two subscripts, j for level 2 units (e.g. subjects) and i 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

$$\nu_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta} + \mathbf{z}^{(2)\prime}_{ij}\boldsymbol{\eta}^{(2)}_j, \tag{7.2}$$

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 $\theta_{ij} = \exp(\nu_{ij})$. 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

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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 response variable, $\log(y_s)$ as an offset and dummies for the time intervals as explanatory variables.

Therefore, one approach to multilevel survival modeling 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 unique failure times sorted in ascending order, each interval starts at (just after) a unique failure time and ends at (just after) the next unique failure time. The units contributing to the likelihood for given intervals then correspond to the 'risk sets' of Cox's proportional 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 from Stata 7).

7.2 Proportional hazards model with random coefficients

We will analyze the dataset published in Danahy *et al.* (1976) and previously analyzed by Pickles and Crouchley (1994; 1995) and Skrondal and Rabe-Hesketh (2004). Here 21 subjects with coronary heart disease participated in a randomized 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 i, condition c (drug versus placebo) within subjects j. 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.

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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

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

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

	Condition $c=1$	Condition $c=2$		
occasion i	Placebo t_{i1j}	ISDN t_{i2j}	x_T	x_D
1	t_{11j}	t_{12j}	0	0
2	t_{21j}	t_{22j}	1	-1
3	t_{31j}	t_{32j}	1	0
4	t_{41j}	t_{42j}	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

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

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

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

where (η_{0j}, η_{1j}) 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:

```
. list subj occ secondp secondi unceni in 1/12, clean

       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 and 0

COBRA
BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 id=_n
. gen after=occ>1
. gen decl=cond(occ>1,occ-3,0)
. sort subj occ
. list id subj occ after decl in 1/12, clean
        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 standardize `secondp` which will be used as a covariate:

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

First we analyze 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
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
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 | 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, clean

        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 labeled in increasing order of the associated survival times and therefore in decreasing order of risk set size (as fewer individuals '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:

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

Hosted by The Berkeley Electronic Press

```
. 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, clean
        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

```
stset secondi, failure(unceni) id(id)
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 approaches 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 regression 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
Iteration 0:   log likelihood = -336.54949
Iteration 1:   log likelihood = -328.60226
Iteration 2:   log likelihood = -328.32219
Iteration 3:   log likelihood = -328.31449
```

COBRA
Collection of Biostatistics
Research Archive

```
Iteration 4:   log likelihood = -328.31446

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)
------------------------------------------------------------------------------
```

Estimates of the fixed effects parameters are fairly 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) f(poiss) offset(lny) adapt eform

Running adaptive quadrature
Iteration 0:    log likelihood = -322.78228
Iteration 1:    log likelihood = -315.21257
Iteration 2:    log likelihood = -315.01199
Iteration 3:    log likelihood = -315.00874
Iteration 4:    log likelihood = -315.00874

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -315.00874
Iteration 1:   log likelihood = -315.00874  (backed up)
Iteration 2:   log likelihood = -315.00822
Iteration 3:   log likelihood = -315.00822

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

Condition Number = 9.3795391

gllamm model

log likelihood = -315.00822

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   4.471778   1.068238     6.27   0.000     2.799893    7.141986
          t2 |   .5237084   .0801868    -4.22   0.000     .3879352    .7070009
          t3 |   1.575047   .1737051     4.12   0.000     1.268873    1.955099
          t4 |   .7716135   .0610165    -3.28   0.001       .66083     .900969
     secondp |   .2328122   .0718344    -4.72   0.000     .1271647    .4262307
       after |   .4108259   .1260568    -2.90   0.004     .2251527    .7496152
        decl |   2.195706   .4598916     3.76   0.000      1.45643    3.310234
         lny |   (offset)
------------------------------------------------------------------------------
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

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


***level 2 (subj)

    var(1): 1.6299871 (.79045789)
------------------------------------------------------------------------------
```

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 12 quadrature points gives quite similar estimates:

```
. matrix a=e(b)

. gllamm unceni t1-t4 secondp after decl, i(subj) nip(12) /*
>   */ eqs(cons) f(poiss) from(a) offset(lny) adapt eform

Running adaptive quadrature
Iteration 0:    log likelihood = -315.00735
Iteration 1:    log likelihood = -315.00732

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -315.00732
Iteration 1:   log likelihood = -315.00732  (backed up)
Iteration 2:   log likelihood = -315.00729
Iteration 3:   log likelihood = -315.00729

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

Condition Number = 9.3879638

gllamm model

log likelihood = -315.00729

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   4.477254   1.076552     6.23   0.000     2.794737    7.172698
          t2 |   .5234166   .0804139    -4.21   0.000     .3873246    .7073265
          t3 |   1.575231   .1737783     4.12   0.000     1.268938    1.955457
          t4 |   .7716079   .0610164    -3.28   0.001     .6608246    .9009634
     secondp |   .2328443   .0717695    -4.73   0.000     .1272624    .4260211
       after |   .4108386   .1260547    -2.90   0.004     .2251661     .749617
        decl |   2.196476   .4603639     3.75   0.000     1.456537    3.312314
         lny |   (offset)
------------------------------------------------------------------------------


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


***level 2 (subj)

    var(1): 1.6372166 (.80952743)
------------------------------------------------------------------------------
```

For presenting final results, it might be worth estimating the model with a few more quadrature points per dimension.

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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

the hazards compared with baseline and that there is a 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 -315.00729))
2.484e-07
```

Note that these models could also be estimated using the `xtpois` command (using ordinary quadrature):

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

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`). We encountered some convergence problems using adaptive quadrature and will therefore get initial estimates using ordinary quadrature and then use these as starting values for adaptive quadrature:

```
. eq after: after
. gllamm unceni t1-t4 secondp after decl, i(subj) nrf(2) eqs(cons after) /*
>     */ family(poiss) offset(lny) eform
Iteration 0:   log likelihood = -320.41886  (not concave)
Iteration 1:   log likelihood =  -308.4702  (not concave)
Iteration 2:   log likelihood = -307.74832
Iteration 3:   log likelihood = -307.71894
Iteration 4:   log likelihood = -307.71848
Iteration 5:   log likelihood = -307.71846

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

Condition Number = 34.59201

gllamm model

log likelihood = -307.71846

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   6.201686   1.901195     5.95   0.000     3.400671     11.3098
          t2 |   .4693898   .0892829    -3.98   0.000     .3233151    .6814613
          t3 |   1.664057   .2156613     3.93   0.000     1.290782    2.145279
          t4 |   .7491887   .0649483    -3.33   0.001     .6321196    .8879391
     secondp |   .1844601   .0850213    -3.67   0.000     .0747423    .4552377
       after |   .2884553   .1260457    -2.85   0.004     .1224986    .6792445
        decl |    2.27134   .4994648     3.73   0.000      1.47606    3.495105
         lny |   (offset)
------------------------------------------------------------------------------
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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


***level 2 (subj)

    var(1): .481472 (.51776169)
    cov(2,1): .74484812 (.82565677) cor(2,1): .74365126

    var(2): 2.0836548 (1.4233089)
------------------------------------------------------------------------------



. matrix a=e(b)

. gllamm unceni t1-t4 secondp after decl, i(subj) nrf(2) eqs(cons after) /*
>     */ family(poiss) offset(lny) nip(12) adapt eform from(a)

Running adaptive quadrature
Iteration 0:    log likelihood = -307.87765
Iteration 1:    log likelihood = -307.83794
Iteration 2:    log likelihood = -307.76703
Iteration 3:    log likelihood = -307.76186
Iteration 4:    log likelihood = -307.75964
Iteration 5:    log likelihood = -307.75964

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -307.75964
Iteration 1:   log likelihood = -307.75964  (backed up)
Iteration 2:   log likelihood = -307.75942
Iteration 3:   log likelihood = -307.75942

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

Condition Number = 45.516143

gllamm model

log likelihood = -307.75942

------------------------------------------------------------------------------
      unceni |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          t1 |   5.794447   1.578055     6.45   0.000     3.397789    9.881607
          t2 |   .4764362   .0847481    -4.17   0.000     .3361968    .6751742
          t3 |   1.661396   .2033098     4.15   0.000     1.307101    2.111724
          t4 |   .7511578   .0644907    -3.33   0.001     .6348208    .8888146
     secondp |     .19056    .063285    -4.99   0.000     .0993911    .3653558
       after |   .3151838   .1305154    -2.79   0.005     .1399869     .709644
        decl |   2.235158   .4842105     3.71   0.000     1.461872    3.417489
         lny |   (offset)
------------------------------------------------------------------------------


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


***level 2 (subj)

    var(1): .37108179 (.41492511)
    cov(2,1): .78380774 (.44013794) cor(2,1): .99999935

    var(2): 1.6555794 (1.0422347)
------------------------------------------------------------------------------
```

The estimates of the covariance matrix are quite different but the fixed effects estimates have not

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

changed substantially. We can check how much the log-likelihood for these parameter estimates changes when 30 quadrature points are used per dimension using the `eval` option:

```
gllamm unceni t1-t4 secondp after decl, i(subj) nrf(2) eqs(cons after) /*
     */ family(poiss) offset(lny) nip(30) adapt eform from(a) eval
```

giving a log-likelihood of -307.75976, very similar to the log-likelihood for 12 points per dimension.

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. Note, however, that the correlation between the random intercept and slope is virtually 1 and hence on the boundary of the parameter space.

Skrondal and Rabe-Hesketh (2004) analyze these data also using survival models with common factors, latent classes and non-parametric maximum likelihood.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 8

Ordinal responses

8.1 Generalizing models for ordinal responses

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

$$y_i^* = \nu_i + \epsilon_i \tag{8.1}$$

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

$$y_i = \begin{cases} a_1 & \text{if } y_i^* \le \kappa_1 \\ a_2 & \text{if } \kappa_1 < y_i^* \le \kappa_2 \\ \vdots & \vdots \\ a_S & \text{if } \kappa_{S-1} < y_i^* \end{cases} \tag{8.2}$$

where the thresholds κ_s do not vary between subjects.

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

$$\mathrm{P}(y_i \le a_s) = F(\kappa_s - \nu_i), \quad s = 0, \ldots, S \tag{8.3}$$

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

$$\begin{aligned} \mathrm{P}(y_i = a_s) &= \mathrm{P}(\kappa_{s-1} < y_i^* \le \kappa_s) \\ &= F(\kappa_s - \nu_i) - F(\kappa_{s-1} - \nu_i). \end{aligned} \tag{8.4}$$

We can equivalently write the model as a cumulative model

$$g(\mathrm{P}(y_i < a_s)) = \kappa_s - \nu_i,$$

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_i of the latent response y_i^* is assumed to have a logistic distribution,

$$\begin{aligned} \Pr(y_i \le a_s) &= \Pr(y_i^* \le \kappa_s) \\ &= \frac{\exp(\kappa_s - \nu_i)}{1 + \exp(\kappa_s - \nu_i)} \end{aligned} \tag{8.5}$$

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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

$$\log\left(\frac{\Pr(y_i \leq a_s)}{1-\Pr(y_i \leq a_s)}\right) = \kappa_s - \nu_i \tag{8.6}$$

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

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

$$\Pr(y_i \leq a_s) = \Phi(\kappa_s - \nu_i) \tag{8.7}$$

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

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

$$\ln(-\ln(1-\Pr(y_i \leq a_s))) = \kappa_s - \nu_i \tag{8.8}$$

or, equivalently,

$$\Pr(y_i \leq a_s) = 1 - \exp(-\exp(\kappa_s - \nu_i)) \tag{8.9}$$

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_{i1} to x_{ip} as

$$\kappa_{si} = \alpha_{s0} + \alpha_{s1}x_{i1} + \cdots + \alpha_{sp}x_{ip}.$$

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

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. When an ordinal probit link is used, the models are known as *graded response models*. Partial credit models can be estimated using the `mlogit` link; see Chapter 9.

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 $\mathrm{Var}(\epsilon)$. Further, the effect of the latent variable η_j is the same for all items and the thresholds κ_s are constant across items i:

$$g(\mathrm{P}(y_{ij} \leq a_s)) = \kappa_s - (\beta_i + \eta_j), \quad \beta_1 = 0 \tag{8.10}$$

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 η_j to differ between items by including factor loadings λ_i in the model

$$g(\mathrm{P}(y_{ij} \le a_s)) = \kappa_s - (\beta_i + \eta_j \lambda_i), \quad \beta_1 = 0,\ \lambda_1 = 1 \tag{8.11}$$

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

$$\mathrm{P}(y_{ij} \le a_s) = \Phi((\kappa_s - \beta_i - \eta_j \lambda_i)/\sigma_i), \quad \beta_1 = 0,\ \lambda_1 = 1,\ \sigma_1 = 1 \tag{8.12}$$

where σ_i 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

$$\begin{aligned} g(\mathrm{P}(y_{ij} \le a_s)) &= (\kappa_s - \beta_i)/\sigma_i - \eta_j \lambda_i/\sigma_i \\ &= \kappa^*_{si} - \eta_j \lambda^*_i \end{aligned}$$

where

$$\kappa^*_{si} = (\kappa_s - \beta_i)/\sigma_i.$$

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:

$$g(\mathrm{P}(y_{ij} \le a_s)) = \kappa_{si} - \eta_j \lambda_i, \quad \beta_1 = 0 \tag{8.13}$$

Finally, covariates can be incorporated in different ways: (1) the covariate affects the latent response indirectly by affecting the latent variable η_j (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 analyzed in Hedeker and Gibbons (1994). Schools were randomized 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.

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

Hosted by The Berkeley Electronic Press

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 dichotomized THKS score.)

The two-level model can be written as

$$\eta_{ijk} = \beta_0 + \beta_1 x_{Pijk} + \beta_2 x_{Ck} + \beta_3 x_{Tk} + \beta_4 x_{Ck} x_{Tk} + \eta_j, \tag{8.14}$$

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

The three level model is

$$\eta_{ijk} = \beta_0 + \beta_1 x_{Pijk} + \beta_2 x_{Ck} + \beta_3 x_{Tk} + \beta_4 x_{Ck} x_{Tk} + \eta^{(2)}_{jk} + \eta^{(3)}_{k}, \tag{8.15}$$

where the response variable is modeled 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
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 (sum) 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:

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
. list school class thk cc tv wt1 in 80/90, clean

       school    class   thk   cc   tv   wt1
 80.      199   199106     1    0    1     1
 81.      405   405101     1    0    1     1
 82.      405   405102     1    0    1     6
 83.      405   405103     1    0    1     1
 84.      407   407101     1    0    1     1
 85.      407   407102     1    0    1     2
 86.      407   407103     1    0    1     2
 87.      506   506103     1    0    1     2
 88.      506   506105     1    0    1     2
 89.      506   506107     1    0    1     1
 90.      506   506110     1    0    1     2
```

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) link(ologit) f(binom) /*
>   */ weight(wt) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -2117.2895
Iteration 1:    log likelihood = -2115.4014
Iteration 2:    log likelihood = -2115.3876
Iteration 3:    log likelihood = -2115.3836
Iteration 4:    log likelihood = -2115.3835

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2115.3835
Iteration 1:   log likelihood = -2115.3835  (backed up)
Iteration 2:   log likelihood = -2115.3831
Iteration 3:   log likelihood = -2115.3831

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

Condition Number = 15.444475

gllamm model

log likelihood = -2115.3831

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
thk          |
      prethk |    .414802   .0393628    10.54   0.000     .3376523    .4919518
          cc |    .861341   .1735825     4.96   0.000     .5211255    1.201556
          tv |   .2058666   .1705965     1.21   0.228    -.1284965    .5402296
        cctv |  -.3011398   .2451323    -1.23   0.219    -.7815902    .1793106
-------------+----------------------------------------------------------------
_cut11       |
       _cons |  -.0757384   .1466273    -0.52   0.605    -.3631227    .2116458
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   1.197664   .1485246     8.06   0.000     .9065614    1.488767
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   2.403179   .1579048    15.22   0.000     2.093691    2.712667
------------------------------------------------------------------------------


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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
***level 2 (class)

    var(1): .18862146 (.06372493)
------------------------------------------------------------------------------
```

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) adapt from(a) skip
```

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

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

```
. matrix a=e(b)
. gllamm thk prethk cc tv cctv, i(class school) link(ologit) /*
>  */ f(binom) weight(wt) from(a) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -2115.3831
Iteration 1:    log likelihood = -2115.3831
Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -2115.3831  (not concave)
Iteration 1:   log likelihood = -2115.3831  (not concave)
Iteration 2:   log likelihood = -2115.3831  (not concave)
Iteration 3:   log likelihood = -2114.8169
Iteration 4:   log likelihood = -2114.5882
Iteration 5:   log likelihood = -2114.5881

number of level 1 units = 1600
number of level 2 units = 135
number of level 3 units = 28

Condition Number = 16.631187

gllamm model

log likelihood = -2114.5881

------------------------------------------------------------------------------
         thk |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
thk          |
      prethk |   .4085277   .0396155    10.31   0.000     .3308826    .4861727
          cc |   .8841594   .2097996     4.21   0.000     .4729596    1.295359
          tv |   .2362118    .204815     1.15   0.249    -.1652182    .6376418
        cctv |  -.3715189   .2956721    -1.26   0.209    -.9510257    .2079878
-------------+----------------------------------------------------------------
_cut11       |
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
       _cons |  -.0961882   .1690349    -0.57   0.569    -.4274904    .2351141
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   1.177237   .1706267     6.90   0.000     .8428148    1.511659
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   2.383431   .1787935    13.33   0.000     2.033002     2.73386
------------------------------------------------------------------------------


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

***level 2 (class)

    var(1): .14821764 (.0637401)

***level 3 (school)

    var(1): .04487641 (.04253446)
------------------------------------------------------------------------------
```

The variance component for schools does not appear to be 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 current parameter estimates using larger numbers of quadrature points by using the `eval` option:

```
. gllamm thk prethk cc tv cctv, i(class school) link(ologit) /*
>  */ f(binom) weight(wt) nip(30) eval from(a) adapt
```

The log-likelihood values do not change at all when more quadrature points are used and the 8-point approximation therefore appears to be adequate.

8.4 Item response models with an explanatory variable

We will analyze six ordinal items relating to delinquency from Udry (1998). The following items were rated as 0: not true, 1: sometimes true and 2: often true:

1. Hangs around kids who get in trouble
2. Cheats or tells lies
3. Bullies or is cruel/mean to others
4. Does not feel sorry after misbehaving
5. Breaks things on purpose
6. Is disobedient at school

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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 patterns for 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`.

```
. qui tab item, gen(d)
. gllamm y d2-d6, i(id) weight(wt) l(oprob) f(binom) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -10742.499
Iteration 1:    log likelihood = -10235.427
Iteration 2:    log likelihood =  -10225.11
Iteration 3:    log likelihood = -10225.108

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10225.108
Iteration 1:   log likelihood = -10225.108  (backed up)
Iteration 2:   log likelihood = -10225.093
Iteration 3:   log likelihood = -10225.093

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

Condition Number = 8.3491714

gllamm model

log likelihood = -10225.093

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          d2 |   .1333404   .0370122     3.60   0.000     .0607979     .205883
          d3 |  -.3568232   .0423794    -8.42   0.000    -.4398853   -.2737611
          d4 |  -.3840764   .0427277    -8.99   0.000    -.4678211   -.3003316
          d5 |  -.5170362   .0447789   -11.55   0.000    -.6048012   -.4292712
          d6 |  -.0324827   .0387613    -0.84   0.402    -.1084535    .0434881
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.997424   .0388143    51.46   0.000     1.921349    2.073499
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.783248   .0445798    62.43   0.000     2.695874    2.870623
------------------------------------------------------------------------------
```

COBRA
BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
_cut13       |
       _cons |    3.13726   .0484972    64.69   0.000     3.042207    3.232312
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): 1.0424215 (.05556149)
------------------------------------------------------------------------------
```

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. Before estimating other models, we will save the estimates using

```
estimates store mod1
```

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

```
. eq load: d1-d6
. gllamm y d2-d6, i(id) weight(wt) l(oprob) f(binom) eqs(load) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -10728.067
Iteration 1:    log likelihood = -10246.429
Iteration 2:    log likelihood = -10167.963
Iteration 3:    log likelihood = -10155.445
Iteration 4:    log likelihood = -10154.595
Iteration 5:    log likelihood =  -10154.59

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood =  -10154.59
Iteration 1:   log likelihood = -10154.422
Iteration 2:   log likelihood =   -10154.4
Iteration 3:   log likelihood =   -10154.4

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

Condition Number = 28.859952

gllamm model

log likelihood = -10154.4

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          d2 |   .0426769   .0589893     0.72   0.469    -.0729399    .1582938
          d3 |  -1.170732   .1329432    -8.81   0.000    -1.431296   -.9101682
          d4 |   -1.01723   .1186416    -8.57   0.000    -1.249763   -.7846963
          d5 |  -.8761092    .104336    -8.40   0.000    -1.080604   -.6716145
          d6 |  -.7456524   .1109123    -6.72   0.000    -.9630365   -.5282683
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   1.701352   .0426219    39.92   0.000     1.617815     1.78489
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   2.497098   .0469756    53.16   0.000     2.405027    2.589169
-------------+----------------------------------------------------------------
_cut13       |
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
       _cons |   2.863077   .0503314    56.88   0.000     2.764429    2.961725
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): .49473593 (.05947006)

    loadings for random effect 1
    d1: 1 (fixed)
    d2: 1.1341268 (.08848816)
    d3: 2.0171066 (.16925857)
    d4: 1.8242372 (.15393464)
    d5: 1.5214631 (.13277401)
    d6: 1.9696146 (.16213068)

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

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

```
. estimates store mod2
. lrtest mod1 mod2
(log-likelihoods of null models cannot be compared)
likelihood-ratio test                                 LR chi2(5)  =    141.39
(Assumption: mod1 nested in mod2)                     Prob > chi2 =    0.0000
```

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,

$$\ln \sigma_{ij} \;=\; \mathbf{z}_{ij}^{(0)\prime}\boldsymbol{\alpha}. \tag{8.16}$$

(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: d1-d6
```

However, we need to constrain the first scale to 1. One way to do this is to simply use

```
eq het: d2-d6
```

because omission of `d1` is equivalent to setting the corresponding coefficient $\ln \sigma_{1j}$ equal to 0. For illustration we also use a more convoluted approach to demonstrate how to use constraints in `gllamm`. First, 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 d2-d6, i(id) weight(wt) l(soprob) f(binom) eqs(load) s(het) /*
> */ trace noest

General model information
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
------------------------------------------------------------------------------
dependent variable:         y
ordinal responses:         soprobit
denominator:                1
equations for fixed effects
                           y:  d2 d3 d4 d5 d6
                           _cut11:  _cons
                           _cut12:  _cons
                           _cut13:  _cons


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

   log standard deviation
   lns1: d1 d2 d3 d4 d5 d6


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


   lambdas for random effect 1
   id1_1l: d2 d3 d4 d5 d6
   standard deviation for random effect 1
   id1_1 : d1
```

Under 'level 1 equation', we are informed that the log standard deviation parameters have equation name **lns1** and column names **d1**, **d2**, 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:

```
. cons def 1 [lns1]d1=1

. gllamm y d2-d6, i(id) weight(wt) l(soprob) f(binom) eqs(load) s(het) /*
>  */ constr(1) adapt

Running adaptive quadrature
Iteration 0:    log likelihood = -13709.163
Iteration 1:    log likelihood = -13596.025
Iteration 2:    log likelihood = -11325.335
Iteration 3:    log likelihood = -10985.475
Iteration 4:    log likelihood = -10590.418
Iteration 5:    log likelihood = -10454.609
Iteration 6:    log likelihood = -10371.479
Iteration 7:    log likelihood = -10234.759
Iteration 8:    log likelihood = -10173.309
Iteration 9:    log likelihood = -10154.377
Iteration 10:    log likelihood =  -10120.13
Iteration 11:    log likelihood = -10111.119
Iteration 12:    log likelihood = -10109.773
Iteration 13:    log likelihood = -10109.715
Iteration 14:    log likelihood = -10109.627
Iteration 15:    log likelihood = -10109.627

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10109.627
Iteration 1:   log likelihood = -10109.627  (backed up)
Iteration 2:   log likelihood = -10109.596
Iteration 3:   log likelihood = -10109.596

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

Condition Number = 55.843988
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

```
gllamm model with constraints:
 ( 1)  [lns1]d1 = 1

log likelihood = -10109.59597676592

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
          d2 |  -.0107539   .3054669    -0.04   0.972     -.609458    .5879501
          d3 |  -3.489128   .6168898    -5.66   0.000     -4.69821   -2.280046
          d4 |  -2.887489   .5721521    -5.05   0.000    -4.008886   -1.766091
          d5 |  -2.420719     .56635    -4.27   0.000    -3.530745   -1.310694
          d6 |  -5.475978   .7422003    -7.38   0.000    -6.930663   -4.021292
-------------+----------------------------------------------------------------
_cut11       |
       _cons |   4.673623   .1225696    38.13   0.000     4.433391    4.913855
-------------+----------------------------------------------------------------
_cut12       |
       _cons |   7.049441    .174879    40.31   0.000     6.706684    7.392197
-------------+----------------------------------------------------------------
_cut13       |
       _cons |   8.183033   .2148669    38.08   0.000     7.761902    8.604165
------------------------------------------------------------------------------

Variance at level 1
------------------------------------------------------------------------------

    equation for log standard deviation:

    d1: 1 (0)
    d2: 1.0254298 (.06087759)
    d3: .94857302 (.0862344)
    d4: .94999093 (.08079316)
    d5: 1.0089726 (.08000006)
    d6: 1.5691117 (.072496)

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


***level 2 (id)

    var(1): 4.0522808 (.50534715)

    loadings for random effect 1
    d1: 1 (fixed)
    d2: 1.158027 (.09592887)
    d3: 2.10297 (.18929612)
    d4: 1.8620151 (.16899761)
    d5: 1.4831383 (.14407552)
    d6: 2.456456 (.22062851)

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

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*(10154.42-10109.60))
7.997e-18
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Item-specific thresholds

First we fit the model in equation (8.13) without factor loadings, or equivalently, with $\lambda_i = 1$. 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) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -10672.359
Iteration 1:    log likelihood = -10166.217
Iteration 2:    log likelihood = -10154.907
Iteration 3:    log likelihood = -10154.882
Iteration 4:    log likelihood = -10154.878

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10154.878
Iteration 1:   log likelihood = -10154.839
Iteration 2:   log likelihood = -10154.804
Iteration 3:   log likelihood = -10154.804

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

Condition Number = 9.8401115

gllamm model

log likelihood = -10154.804
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_cut11						
_cons	1.965004	.0389448	50.46	0.000	1.888673	2.041334
_cut12						
_cons	2.959698	.0581748	50.88	0.000	2.845678	3.073719
_cut13						
_cons	3.381418	.0739307	45.74	0.000	3.236516	3.526319
_cut21						
_cons	1.836902	.037195	49.39	0.000	1.764001	1.909803
_cut22						
_cons	2.76768	.0529078	52.31	0.000	2.663983	2.871378
_cut23						
_cons	3.191754	.0656071	48.65	0.000	3.063167	3.320342
_cut31						
_cons	2.367054	.0456407	51.86	0.000	2.2776	2.456508
_cut32						
_cons	3.087231	.0619507	49.83	0.000	2.96581	3.208652
_cut33						
_cons	3.445663	.0749175	45.99	0.000	3.298827	3.592498
_cut41						
_cons	2.383722	.0459178	51.91	0.000	2.293725	2.473719
_cut42						
_cons	3.155176	.0641031	49.22	0.000	3.029536	3.280816

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

```
--------------+----------------------------------------------------------------
_cut43        |
        _cons |   3.543821   .0800002    44.30   0.000     3.387023    3.700618
--------------+----------------------------------------------------------------
_cut51        |
        _cons |    2.50479   .0482017    51.96   0.000     2.410317    2.599264
--------------+----------------------------------------------------------------
_cut52        |
        _cons |   3.343634   .0718635    46.53   0.000     3.202785    3.484484
--------------+----------------------------------------------------------------
_cut53        |
        _cons |   3.792863   .0955917    39.68   0.000     3.605507     3.98022
--------------+----------------------------------------------------------------
_cut61        |
        _cons |   2.104489   .0411193    51.18   0.000     2.023897    2.185082
--------------+----------------------------------------------------------------
_cut62        |
        _cons |    2.59496   .0491502    52.80   0.000     2.498627    2.691293
--------------+----------------------------------------------------------------
_cut63        |
        _cons |   2.842882   .0545924    52.07   0.000     2.735883    2.949881
-------------------------------------------------------------------------------


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

***level 2 (id)

    var(1): 1.0469495 (.05588611)
------------------------------------------------------------------------------
```

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 $\widehat{\sigma}_6$ 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: d2-d6
```

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.)

```
. quietly gllamm y, i(id) weight(wt) l(oprob) f(binom) adapt
. eq thr: d2-d6
. matrix a=e(b)
. gllamm y, i(id) weight(wt) l(oprob) f(binom) from(a) thresh(thr) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -10413.275
Iteration 1:    log likelihood = -10171.745
Iteration 2:    log likelihood = -10169.376
Iteration 3:    log likelihood = -10155.876
Iteration 4:    log likelihood = -10154.886
Iteration 5:    log likelihood = -10154.858
Iteration 6:    log likelihood = -10154.858

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10154.858  (not concave)
Iteration 1:   log likelihood = -10154.858
```

Collection of Biostatistics Research Archive

```
Iteration 2:   log likelihood =   -10154.8
Iteration 3:   log likelihood =   -10154.8

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

Condition Number = 20.748861

gllamm model

log likelihood = -10154.8

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
          d2 |   -.128102   .0379933    -3.37   0.001    -.2025675   -.0536366
          d3 |   .4020526   .0436782     9.20   0.000     .3164449    .4876604
          d4 |   .4187204   .0438804     9.54   0.000     .3327164    .5047243
          d5 |   .5397898   .0457374    11.80   0.000     .4501462    .6294333
          d6 |   .1394867   .0403424     3.46   0.001      .060417    .2185564
       _cons |   1.965016   .0389465    50.45   0.000     1.888683     2.04135
-------------+----------------------------------------------------------------
_cut12       |
          d2 |  -.1920172   .0637919    -3.01   0.003    -.3170469   -.0669874
          d3 |   .1275351   .0700423     1.82   0.069    -.0097452    .2648154
          d4 |   .1954801    .071906     2.72   0.007     .0545469    .3364133
          d5 |   .3839402   .0784624     4.89   0.000     .2301568    .5377236
          d6 |  -.3647384   .0610158    -5.98   0.000    -.4843271   -.2451497
       _cons |   2.959716   .0581813    50.87   0.000     2.845683    3.073749
-------------+----------------------------------------------------------------
_cut13       |
          d2 |  -.1896587   .0857015    -2.21   0.027    -.3576305   -.0216869
          d3 |   .0642512   .0919546     0.70   0.485    -.1159765     .244479
          d4 |   .1624088   .0961209     1.69   0.091    -.0259847    .3508024
          d5 |   .4114549   .1089196     3.78   0.000     .1979765    .6249334
          d6 |  -.5385325   .0783914    -6.87   0.000    -.6921769   -.3848881
       _cons |   3.381434   .0739413    45.73   0.000     3.236512    3.526357
------------------------------------------------------------------------------


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

***level 2 (id)

    var(1): 1.0470079 (.05589428)
------------------------------------------------------------------------------
. estimates store mod3
```

We now include factor loadings in the model. We can use the estimates from the previous model as starting values. However, new parameters will in this case be set to 0 and this is not a good starting point for factor loadings. We therefore construct a matrix of starting values as follows:

```
matrix a=e(b)
matrix b=a[1,1..18],1,1,1,1,1,a[1,19]
```

The factor loadings occur just before the random effect standard deviation. We could have found this out by first running the command below with the `noest` option.

```
. gllamm y, i(id) weight(wt) l(oprob) f(binom) thresh(thr) eqs(load) adapt /*
>  */ from(b) copy
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
Running adaptive quadrature
Iteration 0:    log likelihood = -10154.807
Iteration 1:    log likelihood = -10129.123
Iteration 2:    log likelihood = -10121.847
Iteration 3:    log likelihood =  -10121.08
Iteration 4:    log likelihood = -10118.434
Iteration 5:    log likelihood = -10116.115
Iteration 6:    log likelihood = -10114.192
Iteration 7:    log likelihood =   -10113.5
Iteration 8:    log likelihood = -10111.694
Iteration 9:    log likelihood = -10111.064
Iteration 10:    log likelihood = -10109.396
Iteration 11:    log likelihood = -10109.225
Iteration 12:    log likelihood = -10109.225

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10109.225  (not concave)
Iteration 1:   log likelihood = -10109.225
Iteration 2:   log likelihood = -10108.732
Iteration 3:   log likelihood = -10108.728
Iteration 4:   log likelihood = -10108.728

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

Condition Number = 53.339512

gllamm model

log likelihood = -10108.728

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
          d2 |  -.0391491   .0629311    -0.62   0.534    -.1624918    .0841936
          d3 |   1.443175   .2027458     7.12   0.000     1.045801     1.84055
          d4 |   1.206354   .1581193     7.63   0.000     .8964455    1.516262
          d5 |   .8680135   .1116367     7.78   0.000     .6492096    1.086817
          d6 |   .3944866   .0831255     4.75   0.000     .2315637    .5574095
       _cons |    1.71894   .0451064    38.11   0.000     1.630533    1.807347
-------------+----------------------------------------------------------------
_cut12       |
          d2 |  -.0706805   .0929776    -0.76   0.447    -.2529132    .1115523
          d3 |   1.465181   .2555381     5.73   0.000     .9643357    1.966027
          d4 |    1.22437   .2032371     6.02   0.000     .8260328    1.622708
          d5 |   .8267152   .1498579     5.52   0.000     .5329991    1.120431
          d6 |  -.0087204   .1070746    -0.08   0.935    -.2185828     .201142
       _cons |    2.60969   .0668285    39.05   0.000     2.478709    2.740672
-------------+----------------------------------------------------------------
_cut13       |
          d2 |   -.056854   .1126775    -0.50   0.614    -.2776978    .1639897
          d3 |   1.540863   .2843446     5.42   0.000     .9835578    2.098168
          d4 |   1.306609   .2305669     5.67   0.000     .8547058    1.758511
          d5 |   .9010973   .1776548     5.07   0.000     .5529003    1.249294
          d6 |  -.1394464   .1223704    -1.14   0.254    -.3792881    .1003952
       _cons |   2.987063    .081527    36.64   0.000     2.827274    3.146853
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): .54879138 (.06846412)

    loadings for random effect 1
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
    d1: 1 (fixed)
    d2: 1.1287193 (.09368312)
    d3: 2.2137771 (.23415021)
    d4: 1.9576643 (.19284791)
    d5: 1.4694014 (.13819925)
    d6: 1.3896627 (.11975133)

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

The likelihood ratio test indicates that factor loadings are required:

```
. estimates store mod4

. lrtest mod3 mod4
(log-likelihoods of null models cannot be compared)

likelihood-ratio test                                  LR chi2(5)  =     92.15
(Assumption: mod3 nested in mod4)                      Prob > chi2 =    0.0000
```

The scaled probit model in (8.13) is nested in this model since the thresholds are freely estimated here whereas the scaled probit model imposes the constraints

$$\kappa_{si} = (\kappa_{s1} - \beta_i)/\sigma_i, \quad i = 2, \ldots, 6$$

for some β_i and positive σ_i. 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.

$$\eta_j = \gamma x_j + \zeta_j,$$

where x_j 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) f(binom) thresh(thr) eqs(load) geqs(f1) f
> rom(a) adapt

Running adaptive quadrature
Iteration 0:    log likelihood =  -10108.74
Iteration 1:    log likelihood =   -10084.6
Iteration 2:    log likelihood = -10080.763
Iteration 3:    log likelihood = -10076.382
Iteration 4:    log likelihood = -10073.956
Iteration 5:    log likelihood = -10073.064
Iteration 6:    log likelihood = -10073.063

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10073.063  (not concave)
Iteration 1:   log likelihood = -10072.721
Iteration 2:   log likelihood = -10072.016
Iteration 3:   log likelihood = -10072.007
Iteration 4:   log likelihood = -10072.007

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

COBRA
Collection of Biostatistics
Research Archive

```
Condition Number = 56.253762

gllamm model

log likelihood = -10072.007

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
          d2 |  -.0807287   .0399511    -2.02   0.043    -.1590314   -.002426
          d3 |   .9657559   .1358567     7.11   0.000     .6994817    1.23203
          d4 |   .8510978   .1095756     7.77   0.000     .6363336   1.065862
          d5 |   .7096138   .0762709     9.30   0.000     .5601255   .8591021
          d6 |   .2738164    .055881     4.90   0.000     .1642917   .3833411
       _cons |   1.330609   .0518214    25.68   0.000     1.229041   1.432177
-------------+----------------------------------------------------------------
_cut12       |
          d2 |  -.1018712    .069341    -1.47   0.142    -.2377771   .0340347
          d3 |   .9903191   .1831365     5.41   0.000     .6313781    1.34926
          d4 |   .8811183   .1520571     5.79   0.000     .5830919   1.179145
          d5 |   .6867659   .1151216     5.97   0.000     .4611318      .9124
          d6 |  -.1095639    .078896    -1.39   0.165    -.2641973   .0450695
       _cons |   2.209185   .0673109    32.82   0.000     2.077258   2.341112
-------------+----------------------------------------------------------------
_cut13       |
          d2 |   -.083249   .0905504    -0.92   0.358    -.2607246   .0942266
          d3 |   1.066115   .2109018     5.06   0.000     .6527547   1.479474
          d4 |    .967104    .179841     5.38   0.000     .6146221   1.319586
          d5 |   .7696019   .1452699     5.30   0.000     .4848781   1.054326
          d6 |   -.231419   .0952459    -2.43   0.015    -.4180976  -.0447404
       _cons |   2.581637   .0802369    32.18   0.000     2.424375   2.738898
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): .4865584 (.06229725)

    loadings for random effect 1
    d1: 1 (fixed)
    d2: 1.169714 (.09761997)
    d3: 2.2703088 (.23290422)
    d4: 2.041305 (.19962495)
    d5: 1.5527931 (.14671162)
    d6: 1.495958 (.12983416)


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

    random effect 1 has 1 covariates:
    sex: -.23750052 (.02980019)
------------------------------------------------------------------------------
```

Girls are on average less delinquent, an effect that is significant according to the likelihood ratio test:

```
. estimates store mod5

. lrtest mod4 mod5
(log-likelihoods of null models cannot be compared)

likelihood-ratio test                                  LR chi2(1)  =     73.44
(Assumption: mod4 nested in mod5)                      Prob > chi2 =    0.0000
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

The model assumes that being a girl affects the responses to the individual items only by affecting overall latent delinquency η_j, i.e., the effect of being a girl on the ith item is to reduce the latent response by $-0.24\lambda_i$. However, it is quite possible that, even for the same overall delinquency level, girls are more less likely to exhibit certain behaviors than boys. For example, there may be a direct effect of being a girl on the first item (hangs around kids who get in trouble) in addition to the indirect effect via the latent variable. The existence of such a direct effect in an item response model is known as item bias or differential item functioning. The direct effect is simply interaction between sex and the dummy variable for the first item:

```
. matrix a=e(b)
. gen sex_d1 = sex*d1
. gllamm y sex_d1, i(id) weight(wt) l(oprob) f(binom) thresh(thr) eqs(load) /*
>  */ geqs(f1) from(a) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -10071.986
Iteration 1:    log likelihood = -10070.654
Iteration 2:    log likelihood = -10056.956
Iteration 3:    log likelihood = -10054.989
Iteration 4:    log likelihood = -10050.528
Iteration 5:    log likelihood =  -10044.35
Iteration 6:    log likelihood = -10044.234
Iteration 7:    log likelihood = -10043.717
Iteration 8:    log likelihood = -10043.444
Iteration 9:    log likelihood = -10043.438

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10043.438  (not concave)
Iteration 1:   log likelihood = -10043.438
Iteration 2:   log likelihood = -10040.524
Iteration 3:   log likelihood = -10040.408
Iteration 4:   log likelihood = -10040.408

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

Condition Number = 71.04524

gllamm model

log likelihood = -10040.408

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y            |
      sex_d1 |    .456135   .0604211     7.55   0.000     .3377119    .5745581
-------------+----------------------------------------------------------------
_cut11       |
          d2 |  -.7985993   .1032255    -7.74   0.000    -1.000918    -.596281
          d3 |   .1153217   .1585415     0.73   0.467    -.1954138    .4260573
          d4 |   .0352298   .1428646     0.25   0.805    -.2447796    .3152393
          d5 |  -.0412977   .1211773    -0.34   0.733    -.2788009    .1962056
          d6 |  -.4670337   .1112572    -4.20   0.000    -.6850938   -.2489737
       _cons |   1.945897   .1009058    19.28   0.000     1.748126    2.143669
-------------+----------------------------------------------------------------
_cut12       |
          d2 |  -.8595046   .1231405    -6.98   0.000    -1.100856   -.6181536
          d3 |    .083517   .1991245     0.42   0.675    -.3067599    .4737938
          d4 |   .0145374   .1793513     0.08   0.935    -.3369847    .3660596
          d5 |  -.1014433   .1523444    -0.67   0.505    -.4000329    .1971462
          d6 |  -.8857908   .1294407    -6.84   0.000     -1.13949   -.6320917
       _cons |   2.862317   .1148872    24.91   0.000     2.637143    3.087492
------------------------------------------------------------------------------
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
_cut13       |
          d2 |  -.8588851   .1388124    -6.19   0.000    -1.130952   -.5868178
          d3 |   .1325459   .2240823     0.59   0.554    -.3066474    .5717392
          d4 |   .0758812   .2041725     0.37   0.710    -.3242896    .4760521
          d5 |  -.0349791   .1778958    -0.20   0.844    -.3836484    .3136902
          d6 |  -1.023863    .142505    -7.18   0.000    -1.303168   -.7445587
       _cons |   3.252445   .1256126    25.89   0.000     3.006248    3.498641
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): .6394474 (.07958483)

    loadings for random effect 1
    d1: 1 (fixed)
    d2: 1.0029064 (.08345634)
    d3: 1.9015024 (.19163644)
    d4: 1.7249596 (.16721599)
    d5: 1.3407662 (.12516631)
    d6: 1.3045971 (.11110781)


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


    random effect 1 has 1 covariates:
    sex: -.33957305 (.03931287)
------------------------------------------------------------------------------
```

For the same delinquency level, girls tend to have higher scores on the first item. This effect is again significant:

```
.  estimates store mod6

. lrtest mod5 mod6
(log-likelihoods of null models cannot be compared)

likelihood-ratio test                                 LR chi2(1)  =     63.20
(Assumption: mod5 nested in mod6)                     Prob > chi2 =    0.0000
```

The difference in mean latent response for item 1 between an average girl and an average boy is

$$-0.340 \times 1 + 0.456$$

Finally, it could be that the effect of being a girl is not the same for each threshold of item 1. 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 modeled by allowing for a different thresholds equations for each item by specifying the `oprob` link six times as we did at the beginning of this section. We can then specify one equation for each response in the `thresh()` option. For the first item, we want the thresholds to depend on `sex`, but for the other items, we do not want the thresholds to depend on covariates. Therefore, we must specify two equations as follows:

```
. eq sex: sex

. eq n:

. gllamm y, i(id) weight(wt) link(oprob oprob oprob oprob oprob oprob) /*
>  */ lv(item) eqs(load) f(binom) geqs(f1) thresh(sex n n n n n) adapt
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
Running adaptive quadrature
Iteration 0:    log likelihood =  -10654.19
Iteration 1:    log likelihood = -10076.486
Iteration 2:    log likelihood = -10054.758
Iteration 3:    log likelihood = -10042.575
Iteration 4:    log likelihood =  -10040.64
Iteration 5:    log likelihood = -10039.825
Iteration 6:    log likelihood =  -10039.62
Iteration 7:    log likelihood = -10039.568
Iteration 8:    log likelihood = -10039.576

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -10039.576
Iteration 1:   log likelihood = -10039.576
Iteration 2:   log likelihood = -10039.529
Iteration 3:   log likelihood = -10039.529

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

Condition Number = 46.057309

gllamm model

log likelihood = -10039.529

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_cut11       |
         sex |   -.446692   .0614692    -7.27   0.000    -.5671694   -.3262145
       _cons |   1.932426   .1021686    18.91   0.000     1.732179    2.132673
-------------+----------------------------------------------------------------
_cut12       |
         sex |  -.4811805   .0962855    -5.00   0.000    -.6698966   -.2924644
       _cons |   2.902696   .1655612    17.53   0.000     2.578202     3.22719
-------------+----------------------------------------------------------------
_cut13       |
         sex |  -.5925504   .1298775    -4.56   0.000    -.8471056   -.3379952
       _cons |   3.470699   .2264967    15.32   0.000     3.026774    3.914624
-------------+----------------------------------------------------------------
_cut21       |
       _cons |   1.148319   .0572613    20.05   0.000     1.036089    1.260549
-------------+----------------------------------------------------------------
_cut22       |
       _cons |   2.003886   .0688781    29.09   0.000     1.868888    2.138885
-------------+----------------------------------------------------------------
_cut23       |
       _cons |   2.394682   .0791718    30.25   0.000     2.239509    2.549856
-------------+----------------------------------------------------------------
_cut31       |
       _cons |   2.063837   .1407128    14.67   0.000     1.788045    2.339629
-------------+----------------------------------------------------------------
_cut32       |
       _cons |   2.948944   .1744749    16.90   0.000      2.60698    3.290909
-------------+----------------------------------------------------------------
_cut33       |
       _cons |   3.388434   .1946716    17.41   0.000     3.006884    3.769983
-------------+----------------------------------------------------------------
_cut41       |
       _cons |   1.982716   .1204901    16.46   0.000      1.74656    2.218872
-------------+----------------------------------------------------------------
_cut42       |
       _cons |   2.878501    .149498    19.25   0.000      2.58549    3.171511
-------------+----------------------------------------------------------------
_cut43       |
       _cons |    3.33006   .1694246    19.66   0.000     2.997994    3.662126
------------------------------------------------------------------------------
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

```
_cut51       |
       _cons |   1.906022   .0901169    21.15   0.000     1.729396    2.082648
-------------+----------------------------------------------------------------
_cut52       |
       _cons |   2.762481   .1154861    23.92   0.000     2.536132    2.988829
-------------+----------------------------------------------------------------
_cut53       |
       _cons |   3.219286    .138095    23.31   0.000     2.948624    3.489947
-------------+----------------------------------------------------------------
_cut61       |
       _cons |   1.480127   .0760863    19.45   0.000     1.331001    1.629253
-------------+----------------------------------------------------------------
_cut62       |
       _cons |   1.977795   .0836873    23.63   0.000     1.813771    2.141819
-------------+----------------------------------------------------------------
_cut63       |
       _cons |   2.229876   .0888658    25.09   0.000     2.055702    2.404049
------------------------------------------------------------------------------


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


***level 2 (id)

    var(1): .63998381 (.07960631)

    loadings for random effect 1
    d1: 1 (fixed)
    d2: 1.002752 (.08340337)
    d3: 1.9024153 (.19194501)
    d4: 1.7244397 (.16725692)
    d5: 1.3406436 (.12513133)
    d6: 1.3042602 (.11105886)


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

    random effect 1 has 1 covariates:
    sex: -.33899689 (.03926522)
------------------------------------------------------------------------------
```

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. A likelihood ratio test gives:

```
. lrtest mod6 .
(log-likelihoods of null models cannot be compared)
likelihood-ratio test                                 LR chi2(2)  =      1.76
(Assumption: mod6 nested in .)                        Prob > chi2 =    0.4153
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Chapter 9

Nominal responses: Polytomous response, discrete choice and rankings

9.1 Multinomial logit model for polytomous or discrete choice data

Let a index the A possible categories of the polytomous 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_i^a, $a = 1, \ldots, A$ so that the multinomial probability of response category f (the probability that f is chosen) for person i is

$$\Pr(f_i) = \frac{\exp(V_i^f)}{\sum_{a=1}^{A} \exp(V_i^a)} \tag{9.1}$$

This probability model can also be derived by assuming that associated with each alternative is an unobserved 'utility' U_i^a (latent response) and that the alternative with the highest utility is selected. Depending on the situation, utility could mean attractiveness, usefulness (voting/purchasing), or cost-effectiveness (clinical treatments) of the alternative. The utility is modeled as

$$U_i^a = V_i^a + \epsilon_i^a. \tag{9.2}$$

Alternative f is selected if

$$U_i^f > U_i^g \text{ for all } g \neq f \tag{9.3}$$

or, equivalently,

$$U_i^f - U_i^g = V_i^f - V_i^g + (\epsilon_i^f - \epsilon_i^g) > 0. \tag{9.4}$$

If the error term ϵ_i^a has an extreme value distribution of type I (Gumbel), then the differences $(\epsilon_i^f - \epsilon_i^g)$ have a logistic distribution and equation (9.1) follows (McFadden, 1973).

For person-specific covariates, a different coefficient vector $\mathbf{g}^a$ is estimated for each alternative except a reference alternative:

$$V_i^a = \mathbf{g}^{a\prime}\mathbf{x}_i \tag{9.5}$$

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

Although the multinomial logit or *polytomous logistic regression* model usually only includes person-specific covariates, we can also include alternative-specific covariates in `gllamm`. This is done by expanding the data so that for each person there is one record for each alternative available to that person (different alternative sets for different persons are possible) and creating a dummy variable for the variable actually selected. Alternative (and person)-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 (persons) 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

$$U_i^a = V_i^a + \epsilon_i^a \tag{9.6}$$

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 $R_i = (r^1, r^2, \ldots, r^A)$ is obtained if

$$U_i^{r^1} > U_i^{r^2} > \ldots > U_i^{r^A} \tag{9.7}$$

and the probability of a ranking R_i is (Luce, 1959)

$$\Pr(R_i) = \frac{\exp(V_i^{r^1})}{\sum_{s=1}^{A} \exp(V_i^{r^s})} \times \frac{\exp(V_i^{r^2})}{\sum_{s=2}^{A} \exp(V_i^{r^s})} \times \ldots \times \frac{\exp(V_i^{r^A})}{\sum_{s=A-1}^{A} \exp(V_i^{r^s})}. \tag{9.8}$$

This probability can be interpreted as arising from a sequential choice process where the subject initially makes a discrete choice among all alternatives. In the second 'stage', a discrete choice is made among all alternatives except the first since this is no longer available. At each

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

subsequent stage, a discrete 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 discrete choice. The likelihood contribution of each alternative set is the same as that of a person in the discrete choice case (with different alternative sets for different 'persons'). 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 rank the modes of transport in order of preference. 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 detail in Skrondal and Rabe-Hesketh (2003b). See also Skrondal and Rabe-Hesketh (2003a) for a less technical account.

9.3 Polytomous response: Multinomial logit with random intercepts

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' behavior is available on 3939 students (i) in 48 schools (j). 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 student i in school j, x_{ij} as well as random intercepts for school, γ_j^a:

$$V_{ij}^a = g_0^a + g_1^a x_{ij} + \gamma_j^a \tag{9.9}$$

where all effects are set to 0 for $a = 1$, i.e. $g_0^1 = g_1^1 = 0$ and $\gamma_j^1 = 0$, making the first alternative the 'reference category'. There are therefore two random effects, γ_j^2 and γ_j^3 for alternative 2 and 3 and these random effects will be assumed to be correlated.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

9.3.1 Data preparation

The data are available as an ASCII file *jspmix.dat*. We read the data using `infile`:

```
infile scy3 id sex stag ravi fry3 tby using jspmix.dat, clear
```

Here `scy3` is the school identifier, `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):

```
. list scy3 sex tby wt1 in 1/12, clean
       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 logistic 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)
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

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, clean

       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
```

COBRA
BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

9.3.2 Parameter estimation

We now use `alt` as the response 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_j^2 and γ_j^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) /*
         */ adapt
```

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 =-1299.6632) 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) adapt
```

to estimate the model with 8 quadrature points (log-likelihood = -1299.6642) and finally, we estimate the model with 12 quadrature points:

```
. 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(12) weight(wt) /*
>          */ from(a) adapt
Running adaptive quadrature
Iteration 0:    log likelihood = -1299.6642
Iteration 1:    log likelihood = -1299.6642

Adaptive quadrature has converged, running Newton-Raphson
Iteration 0:   log likelihood = -1299.6642
Iteration 1:   log likelihood = -1299.6642  (backed up)
```

COBRA
Collection of Biostatistics
Research Archive

```
number of level 1 units = 3939
number of level 2 units = 48

Condition Number = 4.9555184

gllamm model

log likelihood = -1299.6642

------------------------------------------------------------------------------
         alt |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          a2 |   .5961298   .1406765     4.24   0.000     .3204089    .8718508
          a3 |  -.5658695   .1800264    -3.14   0.002    -.9187147   -.2130243
       a2sex |   .5463168   .1455602     3.75   0.000      .261024    .8316097
       a3sex |   1.101733   .1747511     6.30   0.000     .7592275    1.444239
------------------------------------------------------------------------------


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


***level 2 (scy3)

    var(1): .48315158 (.16789726)
    cov(2,1): .54072149 (.1787848) cor(2,1): .90769189

    var(2): .73449179 (.23634624)
------------------------------------------------------------------------------
```

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 maximized 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

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

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 behavior 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 (note that the `gllamm` estimates were obtained using ordinary quadrature because adaptive quadrature had not been implemented at the time). 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 modeled 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

$$V_j^a = g^a + \gamma_j^a \quad a = 1, 2, 3 \tag{9.10}$$

and

$$V^4 = 0. \tag{9.11}$$

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	0.596	0.141
	Boy	0.445	0.112	0.546	0.146
Cat. 3	Cons	-0.631	0.123	-0.566	0.180
	Boy	0.973	0.137	1.102	0.175
	Var(γ_j^2)	0.115	0.054	0.483	0.168
	Var(γ_j^3)	0.189	0.082	0.734	0.236
	Cov(γ_j^3, γ_j^2)	0.019	0.048	0.541	0.179

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

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(γ_j^2)	0.115	0.117	0.022	0.105	0.016
	Var(γ_j^3)	0.189	0.169	0.032	0.161	0.024
	Cov(γ_j^3, γ_j^2)	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(γ_j^2)	0.494	0.519	0.053	0.119	0.017
	Var(γ_j^3)	0.741	0.740	0.075	0.153	0.023
	Cov(γ_j^3, γ_j^2)	0.549	0.573	0.058	0.025	0.015

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

If subjects fall into latent groups or types (e.g. materialistic and post-materialistic), then this can be modeled by assuming that the random effects (γ_j^1,γ_j^2,γ_j^3) take on discrete values (e_{1r},e_{2r},e_{3r}), $r = 1,\ldots,R$ with probabilities π_r.

If `gllamm` is used with the `ip(f)` option, the g^a represent the mean locations and the γ_j^a represent deviations from the mean. If we use the `ip(fn)` option instead, the γ_j^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, clean

       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 from 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 survival 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)
recode chosen .=0
```

COBRA
Collection of Biostatistics
Research Archive

The data now look like this:

```
. sort patt set item

. list patt set rank item chosen wt2 in 1/20, clean

       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
```

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_0^a and a random intercept γ_j^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) /*
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

```
>     */ ip(f)
Iteration 0:   log likelihood = -6460.8177  (not concave)
Iteration 1:   log likelihood = -6347.5802
Iteration 2:   log likelihood =  -6341.896  (not concave)
Iteration 3:   log likelihood = -6312.7116
Iteration 4:   log likelihood = -6311.6967
Iteration 5:   log likelihood = -6311.6859
Iteration 6:   log likelihood = -6311.6859

number of level 1 units = 20358
number of level 2 units = 2262

Condition Number = 10.249546

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)

    loc1: -2.2346, .57995
  var(1): 1.2959723

    loc2: .18121, -.04703
cov(2,1): -.10509491
  var(2): .00852251

    loc3: -1.6514, .42859
cov(3,1): .957736
cov(3,2): -.07766615
  var(3): .70777613
    prob: 0.2061, 0.7939
------------------------------------------------------------------------------
```

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:

```
. save junk, replace
(note: file junk.dta not found)
file junk.dta saved

. infile item1 item2 item3 item4 wt2 using materia.dat, clear
(24 observations read)

. qui summ wt2

. disp r(sum)
```

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

```
2262
. gen l=wt2*ln(wt2/2262)
. qui summ l
. disp r(sum)
-6269.5248
. use junk, clear
```

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)
Iteration 0:   log likelihood = -7119.4687  (not concave)
Iteration 1:   log likelihood =  -6329.073
Iteration 2:   log likelihood = -6317.0349
Iteration 3:   log likelihood = -6313.9794
Iteration 4:   log likelihood = -6311.7965
Iteration 5:   log likelihood = -6311.6862
Iteration 6:   log likelihood = -6311.6859

number of level 1 units = 20358
number of level 2 units = 2262

Condition Number = 9.3329556

gllamm model

log likelihood = -6311.6859

No fixed effects


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


***level 2 (patt)

    loc1: -.87172, 1.9428
  var(1): 1.295966

    loc2: .43736, .20912
cov(2,1): -.10509343
  var(2): .00852231

    loc3: -.2117, 1.8683
cov(3,1): .9577329
cov(3,2): -.07766519
  var(3): .70777501
    prob: 0.2061, 0.7939
------------------------------------------------------------------------------
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

The log-likelihood is as before but the locations are not centered 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) gateaux(-5 5 20) lf0(`k' `ll')
..............................................................................
> ............................................................................
> ............................................................................
> ............................................................................
> ............................................................................
> ............................................................................
> ............................................................................
>>>> Some dots omitted
maximum gateaux derivative is 3.6234362
Iteration 0:   log likelihood = -6312.8415  (not concave)
Iteration 1:   log likelihood = -6301.0572  (not concave)
Iteration 2:   log likelihood = -6295.2365  (not concave)
Iteration 3:   log likelihood = -6294.8282  (not concave)
Iteration 4:   log likelihood = -6292.5033  (not concave)
Iteration 5:   log likelihood = -6287.7128  (not concave)
Iteration 6:   log likelihood = -6286.4001  (not concave)
Iteration 7:   log likelihood = -6283.9777
Iteration 8:   log likelihood = -6283.6958
Iteration 9:   log likelihood = -6281.4594
Iteration 10:  log likelihood = -6281.3648
Iteration 11:  log likelihood = -6281.3611
Iteration 12:  log likelihood = -6281.3611

number of level 1 units = 20358
number of level 2 units = 2262

Condition Number = 11.369394

gllamm model

log likelihood = -6281.3611

No fixed effects


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


***level 2 (patt)

    loc1: -.76144, 3.1448, 1.8385
  var(1): 2.0418196

    loc2: .55538, .21004, .17141
cov(2,1): -.19209478
  var(2): .02389374

    loc3: -.08916, 1.1827, 2.9561
cov(3,1): .83004279
cov(3,2): -.16114665
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
    var(3): 1.5224162
      prob: 0.225, 0.3211, 0.4538
------------------------------------------------------------------------------
```

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 parametrization. Instead of fixing $\gamma_j^4 = 0$, he uses the constraint $\sum_a \gamma_j^a = 0$. 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.

Skrondal and Rabe-Hesketh (2004) describe a random coefficient multinomial logit model for a conjoint experiment in marketing research and Skrondal and Rabe-Hesketh (2003ab) describe multilevel models for discrete choices and rankings of political parties. Data and do-files for both examples are available at http://www.gllamm.org/examples.html

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

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 (2004) 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

COBRA A BEPRESS REPOSITORY Collection of Biostatistics Research Archive

Hosted by The Berkeley Electronic Press

"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(d)
```

This creates variables `d1`, `d2`, 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

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
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`, `lincom` and `nlcom`. 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). `nlcom` can be used to obtain confidence intervals for nonlinear functions of coefficients.

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 `result`j 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, clean

       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, clean

       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
```

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

(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, clean
       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, clean
       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:

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

```
. gen cons=1
. collapse (sum) wt1=cons, by(ind result)
. gen occpat = _n
. list ind result occpat wt1, clean
       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.

COBRA
Collection of Biostatistics
Research Archive

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

References

Albert, P. S., & Follmann, D. A. 2000. Modeling repeated count data subject to informative dropout. *Biometrics*, **56**, 667–677.

Arulampalam, W., Naylor, A. R., & Smith, J. P. 2004. A Hazard Model of the Probability of Medical School Dropout in the UK. *Journal of the Royal Statistical Society, Series A*, **167**, 157–178.

Baker, D., & Hann, M. 2001. General practitioner services in primary care groups in England: Is there inequity between service availability and population need? *Health & Place*, **7**, 67–74.

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.

Boylan, R. T. 2004. Do the Sentencing Guidelines Influence the Retirement Decisions of Federal Judges? *The Journal of Legal Studies*, **33**, 231–253.

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 dispersion. *Biometrika*, **82**, 81–91.

Campbell, S. M., Hann, M., Hacker, J., Burns, C., Oliver, D., Thapar, A., Mead, N., Safran, D. Gelb, & Roland, M. O. 2001. Identifying predictors of high quality care in English general practice: observational study. *British Medical Journal*, **323**, 784–787.

Clayton, D. G. 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., Feger, H., & Klauer, K. C. (eds), *New Developments in Psychological Choice Modeling.* Amsterdam: 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.

COBRA
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

Danahy, D. T., Burwell, D. T., Aranov, W. S., & Prakash, R. 1976. Sustained hemodynamic and antianginal effect of high dose oral isosorbide dinitrate. *Circulation*, **55**, 381–387.

Daucourt, V., Saillour-Glenisson, F., Michel, P., Jutand, M. A., & Abouelfath, A. 2003. A multicenter cluster randomized controlled trial of strategies to improve thyroid function testing. *Medical Care*, **41**, 432–441.

Davies, R. B. 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. B., & Pickles, A. 1987. A joint trip timing store-type choice model for grocery shopping, including inventory effects and nonparametric control for omitted variables. *Transportation Research A*, **21**, 345–361.

Dayton, C. M., & MacReady, G. B. 1988. Concomitant variable latent class models. *Journal of the American Statistical Association*, **83**, 173–178.

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. *Preventive Veterinary Medicine*, **50**, 127–144.

Dunn, G., Everitt, B. S., & Pickles, A. 1993. *Modelling Covariances and Latent Variables using EQS.* London: Chapman & Hall.

Ebel, B. E., Koepsell, T. D., Bennett, E. E., & Rivara, F. P. 2003. Use of child booster seats in motor vehicles following a community campaign - A controlled trial. *Journal of the American Medical Association*, **289**, 879–884.

Finkelstein, M. A. 2002. A latent variable model for the analysis of variability in the classification of radiographs of pneumoconioses. *Annals of Occupational Hygiene*, **46**, 247–250.

Flay, B. R., & Johnson et al., C. A. 1989. The television, school and family smoking cessation and prevention project: I theoretical basis and program development. *Preventive Medicine*, **17**, 585–607.

Follmann, D. A., & Lambert, D. 1989. Generalizing logistic regression by nonparametric mixing. *Journal of the American Statistical Association*, **84**, 295–300.

Formann, A. K. 1992. Linear logistic latent class analysis for polytomous data. *Journal of the American Statistical Association*, **87**, 476–486.

Gibbons, R. D., & Hedeker, D. 1997. Random effects probit and logistic regression models for three-level data. *Biometrics*, **53**, 1527–1537.

Glance, L. G., Dick, A. W., Osler, T. M., & Mukamel, D. 2003. Using hierarchical modeling to measure ICU quality. *Intensive Care Medicine*, **29**, 2223–2229.

Goldstein, H. 2003. *Multilevel Statistical Models (Third Edition).* London: Arnold.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Goldstein, H., Rasbash, J., Plewis, I., Draper, D., Browne, W. J., 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., Pitblado, J., & Sribney, W. 2003. *Maximum Likelihood Estimation with Stata*. College Station, TX: Stata Press.

Grilli, L., & Rampichini, C. 2003. Alternative specifications of bivariate multilevel probit ordinal response models. *Journal of Educational and Behavioral Statistics*, **28**, 31–44.

Hardouin, J.-N, & Mesbah, M. 2004. Clustering binary variables in subscales using an extended Rasch model and Akaike Information Criterion. *Communication in Statistics Theory and methods*, **33**, 1277–1294.

Heckman, J. J., & Singer, B. 1984. A method of minimizing the impact of distributional assumptions in econometric models for duration data. *Econometrica*, **52**, 271–320.

Hedeker, D., & Gibbons, R. D. 1996. MIXOR: A computer program for mixed-effects ordinal probit and logistic regression analysis. *Computer Methods and Programs in Biomedicine*, **49**, 157–76.

Hedeker, D., Gibbons, R. D., & Flay, B. R. 1994. Random-effects regression models for clustered data: With an example from smoking research. *Journal of Consulting and Clinical Psychology*, **62**, 757–765.

Holmås, T. H. 2002. Keeping nurses at work: A duration analysis. *Health Economics*, **11**, 493–503.

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.

Kaufman, J. S., Dole, N., Savitz, D. A., & Herring, A. H. 2003. Modeling community-level influences on preterm birth among African-American and white women in central North Carolina. *Annals of Epidemiology*, **13**, 377–384.

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**, 29–42.

Lesaffre, E., & Spiessens, B. 2001. On the effect of the number of quadrature points in a logistic random-effects model: An example. *Journal of the Royal Statistical Society, Series C*, **50**, 325–335.

Lin, X., & Breslow, N. E. 1996. Bias correction in generalized linear mixed models with multiple components of dispersion. *Journal of the American Statistical Association*, **91**, 1007–1016.

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.

COBRA
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

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 nonparametric estimate of a mixing distribution using mixtures of Gaussians. *Journal of the American Statistical Association*, **11**, 86–94.

Marshall, M., Lockwood, A., Green, G., Zajac-Roles, G., Roberts, C., & Harrison, G. 2004. Systematic assessments of need and care planning in severe mental illness: Cluster randomised controlled trial. *British Journal of Psychiatry*, **185**, 163–168.

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 (Second Edition).* London: Chapman & Hall.

McCulloch, C. E., & Searle, S. R. 2001. *Generalized, Linear and Mixed Models.* New York: Wiley.

McFadden, D. L. 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. *Journal of the Royal Statistical Society, Series C*, **31**, 214–225.

Pagani, L., & Seghieri, C. 2002. A statistical analysis of teaching effectiveness from students' point of view. *Pages 197–208 of: Developments in Statistics.* Metodoloski Zvezki, vol. 17.

Panageas, K. S., Schrag, D., Riedel, E., Bach, P. B., & Begg, C. B. 2003. The effect of clustering of outcomes on the association of procedure volume and surgical outcomes. *Annals of Internal Medicine*, **139**, 658–665.

Pickles, A., & Crouchley, R. 1994. Generalizations and applications of frailty models for survival and event data. *Statistical Methods in Medical Research*, **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. 2004. *Handbook of Statistical Analyses using Stata (Third Edition).* Boca Raton, FL: Chapman & Hall/CRC.

COBRA
Collection of Biostatistics
Research Archive

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., & Skrondal, A. 2001. Parameterization of multivariate random effects models for categorical data. *Biometrics*, **57**, 1256–1264.

Rabe-Hesketh, S., Pickles, A., & Taylor, C. 2000. sg129: Generalized linear latent and mixed models. *Stata Technical Bulletin*, **53**, 47–57.

Rabe-Hesketh, S., Pickles, A., & Skrondal, A. 2001a. 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. 2001b. 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. 2001c. Multilevel models for censored and latent responses. *Statistical Methods in Medical Research*, **10**, 409–427.

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

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

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. 2003b. Maximum likelihood estimation of generalized linear models with covariate measurement error. *The Stata Journal*, **3**, 386–411.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. 2004a. Generalized multilevel structural equation modeling. *Psychometrika*, **69**, 167–190.

Rabe-Hesketh, S., Pickles, A., & Skrondal, A. 2004b. *Multilevel and Structural Equation Modeling of Continuous, Categorical and Event Data.* College Station, TX: Stata Press.

Rabe-Hesketh, S., Skrondal, A., & Pickles, A. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics*, in press.

Raudenbush, S. W., & Bryk, A. S. 2002. *Hierarchical Linear Models.* Thousand Oaks, CA: Sage.

Rodriguez, G., & 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.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

Skrondal, A., & Rabe-Hesketh, S. 2003a. Generalized linear mixed models for nominal data. *American Statistical Association, Proceedings of the Biometrics Section*, 3931–3936.

Skrondal, A., & Rabe-Hesketh, S. 2003b. Multilevel logistic regression for polytomous data and rankings. *Psychometrika*, **68**, 267–287.

Skrondal, A., & Rabe-Hesketh, S. 2003c. Some applications of generalized linear latent and mixed models in epidemiology: Repeated measures, measurement error and multilevel modelling. *Norwegian Journal of Epidemiology*, **13**, 265–278.

Skrondal, A., & Rabe-Hesketh, S. 2004. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models.* Boca Raton, FL: Chapman & Hall/CRC.

Snijders, T. A. B., & Bosker, R. J. 1999. *Multilevel Analysis.* London: Sage.

Stata Base Reference Manuals. 2003. *Stata Base Reference Manuals.* College Station, TX.

StataCorp. 2003. *Stata Statistical Software: Release 8.0.* 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.*

Udry, J. R. 1998. *National Longitudinal Study of Adolescent Health, Waves I & II, 1994-1996.* Chapel Hill, NC: Carolina Population Center.

Vincent, J.-L., Wilkes, M. M., & Navickis, R. J. 2003. Safety of human albumin–serious averse events reported worldwide in 1998-2000. *British Journal of Anaesthesia*, **91**, 625–630.

Woodhouse, G., Rasbash, J., Goldstein, H., & Yang, M. 1995. Some covariance models for longitudinal count data with overdispersion. *Pages 9–57 of:* Woodhouse, G. (ed), *A Guide to MLn for New Users.* London: Institute of Education.

Yang, M., Rasbash, J., Goldstein, H., & Barbosa, M. 1999. *MLwiN Macros for Advanced Multilevel Modelling.* University of Education, London: Multilevel Models Project.

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Index

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive

Hosted by The Berkeley Electronic Press

COBRA
A BEPRESS REPOSITORY
Collection of Biostatistics
Research Archive