gllamm companion

for

Rabe-Hesketh, S. and Skrondal, A. (2012). *Multilevel and Longitudinal Modeling Using Stata (3rd Edition). Volume I: Continuous Responses.* College Station, TX: Stata Press.

Contents

Preface

The `gllamm` companion shows how most of the models discussed in Volume I of *Multilevel and Longitudinal Modeling Using Stata (3rd Edition)* by Rabe-Hesketh and Skrondal (2012) can be fit using the `gllamm` software.

`gllamm` is a user-written program for maximum likelihood estimation of multilevel and latent variable modeling. The software can estimate models for many different response types, including continuous, binary, ordinal and nominal responses, counts, and discrete-time and continuous-time survival. For all except continuous responses, the likelihood involves integrals that cannot be solved analytically, and `gllamm` therefore uses numerical integration by adaptive quadrature (Rabe-Hesketh et al. 2002, 2005). The use of `gllamm` for estimating multilevel models for non-continuous responses is described in detail in Volume II of *Multilevel and Longitudinal Modeling Using Stata (3rd Edition)* by Rabe-Hesketh and Skrondal (2012). See also the `gllamm` web site (http://www.gllamm.org) for many resources for learning about `gllamm`, including the `gllamm` manual (Rabe-Hesketh et al. 2004a), a tutorial, and worked examples using `gllamm`.

If you use `gllamm` for a publication, the best citation to use is

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

This paper describes the estimation method in detail and evaluates it using simulations.

For continuous responses (assuming normality of the random effects and of the responses given the random effects), the likelihood has a relatively simple closed form, and numerical integration is not necessary. The official Stata programs `xtreg` and `xtmixed` for multilevel modeling of continuous responses exploit the simple form of the likelihood and are therefore considerably faster than `gllamm`. These official Stata programs are also more accurate for continuous responses, although `gllamm` will produce very similar estimates as long as adaptive quadrature is used with a sufficient number of quadrature points (see Rabe-Hesketh and Skrondal 2012b, sec. 10.11). To make sure that enough quadrature points are used, the model should be estimated with increasing numbers of quadrature points until the estimates do not change appreciably as the number of quadrature points is increased. This method for checking accuracy is not demonstrated in the `gllamm` companion, but we do compare the estimates with those from `xtmixed`.

Sometimes `gllamm` is used for models with continuous responses because it has features that none of the official Stata programs provide. Before the release of Stata 11, these features included robust standard errors based on the sandwich estimator (Rabe-Hesketh and Skrondal 2006), inverse probability weighting (Rabe-Hesketh and Skrondal 2006), and heteroscedastic level-1 variances (see Rabe-Hesketh and Skrondal 2012a, sec. 6.4.2). These features are now all available in `xtmixed`. For this reason, we are no longer describing the use of `gllamm` for continuous responses (Volume I) in the third edition of our book. However, the use of `gllamm` for other response types is described in detail in Volume II which also provides a detailed description of the syntax for `gllamm` and its post-estimation commands `gllapred` for prediction and `gllasim` for simulation in Appendices B, C, and D, respectively. Appendix A describes the "bare essentials" of the `gllamm`, `eq`, and `gllapred` commands.

Remaining features of gllamm that are not available in any of Stata's official programs and may sometimes be a reason for using `gllamm` for continuous responses include:

- Fitting models in the Generalized Linear Latent and Mixed (GLLAMM) framework (Rabe-Hesketh et al. 2004b; Skrondal and Rabe-Hesketh 2004, 2007) that which extends linear mixed models by including (among other things)
  - Factor loadings, i.e., parameters that multiply the random effects – see `eqs()` option
  - Regressions among random effects – see `bmatrix()` option
- Modeling the level-1 standard deviation as a function of continuous covariates (via a log-linear model – see `s()` option)
- Imposing linear constraints on the model parameters – see `constraints()` option
- Specifying discrete distributions for the random effects, or leaving the random-effects distributions unspecified using Nonparametric Maximum Likelihood Estimation (NPMLE; Rabe-Hesketh et al. 2003) – see `ip()` option
- Avoiding boundary estimates for random-effects covariance matrices, such as zero variances and perfect correlations, by using Bayes modal estimation (Chung et al. 2011)
- Obtaining certain kinds of predictions (Skrondal and Rabe-Hesketh 2009):
  - Posterior correlations among random effects using the `gllapred` command – see `corr` option
  - Standardized level-2 residuals using the `gllapred` command – see `ustd()` option
  - Log-likelihood contributions from the highest-level clusters

This companion is intended for people wishing to learn how to use `gllamm`. A good way to achieve this is often by starting with simple models, such as linear random intercept models, and then gradually extending the models. The fact that these models can also be estimated using `xtmixed` can be seen as an advantage since it allows you to compare estimates and therefore be confident that you are fitting the model you intend to fit. In the `gllamm` companion, we therefore demonstrate and explain how many of the examples of Volume I of *Multilevel and Longitudinal Modeling Using Stata (3rd Edition)* can be estimated using `gllamm`. We do not use the `link()` and `family()` options for `gllamm` in this companion because linear models with `link(identity)` and `family(gaussian)` are the default.

The do file for this companion can be downloaded from

http://www.gllamm.org/gllamm_companion.do

In this companion, we use the same chapter and section numbers as the book for sections where we want to demonstrate the use of `gllamm` or associated post-estimation commands. Where there are separate subsections in the book for analyses using `xtreg` and `xtmixed`, the companion will introduce a new subsection for `gllamm`. For example, if the book describes the use of `xtreg` in Section 2.5.2 and the use of `xtmixed` in Section 2.5.3, then Section 2.5.4 of this companion will describe the use of `gllamm` for the same example. If the book describes only one command for a given example (typically `xtmixed`), we use the same section number and insert a "(g)" before the section heading to indicate that this is the `gllamm` version of the section. We do not describe the datasets or interpret the estimates in this companion to avoid duplicating material from the book.

Since gllamm is a user-written command, it may not be installed on your computer. You can check by typing

```
. which gllamm
```

If the following message appears,

```
command gllamm not found as either built-in or ado-file
```

install gllamm (assuming that your computer is connected to the internet) by using the ssc command:

```
. ssc install gllamm
```

Occasionally, you should update gllamm using ssc with the replace option:

```
. ssc install gllamm, replace
```

Chapter 2

Variance components models

2.5 Estimation using Stata

2.5.4 Using gllamm

We now introduce the `gllamm` command, which will be used extensively for models with categorical or discrete responses in later chapters.

The basic `gllamm` command for fitting the variance-components model is:

```
gllamm wm, i(id)
```

Here the fixed part of the model is specified as in `xtreg` and `xtmixed` by listing the response variable `wm` followed by the explanatory variables (here there are no explanatory variables). The `i()` option specifies the cluster identifier, and by default a random intercept is included.

As mentioned in the preface, `gllamm` uses numerical integration, so we add two options to ensure accurate estimates: the `nip(12)` option to use 12 integration points instead of the default of 8 and `adapt` to use adaptive quadrature instead of the default ordinary Gauss-Hermite quadrature. The command and output become:

(*Continued on next page*)

```
. gllamm wm, i(id) nip(12) adapt
number of level 1 units = 34
number of level 2 units = 17

Condition Number = 152.64775

gllamm model

log likelihood = -184.57839

------------------------------------------------------------------------------
          wm |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   453.9116   26.18394    17.34   0.000      402.592    505.2312
------------------------------------------------------------------------------

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

  396.70876 (136.11608)

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


***level 2 (id)

    var(1): 11456.828 (3997.7689)
------------------------------------------------------------------------------
```

The output from `gllamm` first shows the number of units at each level, here 34 units at level 1 (the total number of measurements) and 17 units at level 2 (the subjects). If the `Condition Number` is very large, the model may not be well identified, but here it is not alarming.

Next the maximized log likelihood is given as -184.58 followed by a regression table giving the estimated fixed regression coefficient $\widehat{\beta}$ next to `_cons`.

Estimates and standard errors for the random part of the model are given under the headings "`Variance at level 1`" for the variance θ of the level-1 residuals ϵ_{ij} and "`Variances and covariances of random effects`" and "`***level 2 (id)`" for the variance ψ of the random intercept ζ_j.

`xtreg` and `xtmixed` display the estimated standard deviations instead of variances. We can convert these standard deviations to variances $\widehat{\theta} = 19.91083^2 = 396.44115$ and $\widehat{\psi} = 107.0464^2 = 11,458.932$, which differ slightly from the estimates using `gllamm`. The reason for the discrepancy is that `gllamm` uses numerical integration, whereas `xtreg` and `xtmixed` exploit the closed form of the likelihood for random-effects models with normally distributed continuous responses. The accuracy of the `gllamm` estimates can be improved by increasing the number of integration points (see section 10.11) using the `nip()` option.

Before doing further analyses, we save the `gllamm` estimates using `estimates store`:

```
. estimates store RI
```

We can then compare this model with other models using for example likelihood-ratio tests. We can also `restore` these estimates later without having to refit the model. This will be useful in section 2.11, where we use `gllamm`'s prediction command `gllapred`. Storing estimates means that they remain available during a Stata session; if we require the estimates again in a later Stata session, we can save them in a file using `estimates save` *filename* (a command introduced in Stata release 10).

2.6 Hypothesis tests and confidence intervals

2.6.2 Hypothesis tests and confidence intervals for the between-cluster variance

Likelihood-ratio test

We can perform a likelihood-ratio test for the null hypothesis that the between-cluster variance ψ is zero:

```
. quietly quietly gllamm wm, i(id) init
. lrtest RI .
Likelihood-ratio test                                 LR chi2(1)  =     46.27
(Assumption: . nested in RI)                          Prob > chi2 =    0.0000
```

Here we used the `init` option (for "initial values") which causes `gllamm` to fit the model without any random effects.

As explained on page 88-89 of the book, the p-value is conservative and should be divided by 2.

2.9 Crossed versus nested effects

We can allow the mean of the response variable to depend on the measurement occasion by including a dummy variable for occasion 2 in the model:

```
. generate occ2 = occasion==2
```

(*Continued on next page*)

```
. gllamm wm occ2, i(id) nip(12) adapt

number of level 1 units = 34
number of level 2 units = 17

Condition Number = 154.075

gllamm model

log likelihood = -184.48885
```

wm	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
occ2	2.88257	6.793615	0.42	0.671	-10.43267	16.19781
_cons	452.4716	26.40495	17.14	0.000	400.7188	504.2244

```
Variance at level 1
-----------------------------------------------------------------------------
  392.30228 (134.5612)

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

***level 2 (id)

    var(1): 11460.466 (3998.611)
-----------------------------------------------------------------------------
```

Note that `gllamm` does not yet allow the use of "factor variables" or `fvvarlist` (introduced in Stata 11) to declare explanatory variables as categorical and specify interactions within the estimation command. We must either create dummy variables and products of variables for interactions before running cngllamm or use the prefix command `xi:` to make `gllamm` accept terms of the form `i.`*varname* and `i.`*varname1*`*i.`*varname2* (see [R]xi).

2.11 Assigning values to the random intercepts

2.11.2 Empirical Bayes prediction

We first restore the estimates of the random-intercept model without the occasion 2 dummy variable as a covariate:

```
. estimates restore RI
(results RI are active now)
```

Empirical Bayes predictions can be obtained by using the post-estimation command `gllapred` for `gllamm` with the `u` option. For a two-level random-intercept model, the `gllapred` command produces two variables, one for the posterior means (empirical Bayes predictions) and one for the posterior standard deviations. In the `gllapred` command, the stub of the variable names is specified, here `eb`, and `gllapred` produces the variable `ebm1` for the posterior means and `ebs1` for the posterior standard deviations as stated in the output below:

(*Continued on next page*)

```
. gllapred eb, u
(means and standard deviations will be stored in ebm1 ebs1)
Non-adaptive log-likelihood: -202.25846
 -245.1480  -225.1857  -211.3252  -199.5193  -190.8173  -186.2250
 -184.7457  -184.5784  -184.5784
log-likelihood:-184.57839
```

Adaptive quadrature is used to evaluate the posterior means and standard deviations. This requires several iterations, with each iteration resulting in an improved evaluation of the log-likelihood. The final value of the log likelihood should be the same as that obtained when the model was fit.

We now display the empirical Bayes predictions (see page 113 of the book):

```
. sort id

. format ebm1 %8.2f

. list id ebm1 if occasion==1, clean noobs

    id      ebm1
     1     63.49
     2    -30.88
     3     59.07
     4    -17.61
     5     45.30
     6    155.89
     7    -41.20
     8    -67.74
     9    192.75
    10    -15.15
    11    -27.44
    12    158.84
    13   -206.83
    14     17.78
    15   -187.17
    16    -92.31
    17     -6.79
```

2.11.3 Empirical Bayes standard errors

Comparative standard errors

The comparative standard errors are just the posterior standard deviations which were produced by the `gllapred` command with the `u` option and stored in the variable `ebs1`:

```
. display ebs1[1]
13.963476
```

Diagnostic standard errors

To obtain diagnostic standard errors, we need to subtract the posterior variance from the prior variance (the estimated random-effects variance). The "estimation metric" used by `gllamm` for random-intercept variances is the square root of the variance (which can be positive or negative). By listing the parameter estimates, we can see what equation name and column name `gllamm` uses for the square root of the random-intercept variance:

(*Continued on next page*)

```
. matrix list e(b)

e(b)[1,3]
           wm:        lns1:         id1:
         _cons        _cons        _cons
y1   453.91159    2.9916012    107.03657
```

The diagnostic standard errors are therefore obtained using:

```
. generate diag_SE = sqrt(([id1]_cons)^2 - ebs1^2)

. display diag_SE[1]
106.12186
```

Chapter 3

Random-intercept models with covariates

3.4 Estimation using Stata

3.4.3 Using gllamm

The `gllamm` command for fitting the model by adaptive quadrature with the default of 8 quadrature points is:

```
. gllamm birwt smoke male mage hsgrad somecoll collgrad
>   married black kessner2 kessner3 novisit pretri2 pretri3,
>   i(momid) adapt
number of level 1 units = 8604
number of level 2 units = 3978
Condition Number = 8059.0646
gllamm model
log likelihood = -65145.752
```

birwt	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
smoke	-218.3289	18.20989	-11.99	0.000	-254.0197	-182.6382
male	120.9375	9.558726	12.65	0.000	102.2027	139.6722
mage	8.100549	1.347267	6.01	0.000	5.459954	10.74114
hsgrad	56.84715	25.0354	2.27	0.023	7.778662	105.9156
somecoll	80.68607	27.30916	2.95	0.003	27.1611	134.211
collgrad	90.83273	27.996	3.24	0.001	35.96157	145.7039
married	49.9202	25.50322	1.96	0.050	-.065183	99.90559
black	-211.4138	28.27821	-7.48	0.000	-266.8381	-155.9895
kessner2	-92.91883	19.92625	-4.66	0.000	-131.9736	-53.86409
kessner3	-150.8759	40.83416	-3.69	0.000	-230.9094	-70.84241
novisit	-30.03035	65.69217	-0.46	0.648	-158.7846	98.72394
pretri2	92.8579	23.1926	4.00	0.000	47.40125	138.3145
pretri3	178.7295	51.64148	3.46	0.001	77.5141	279.945
_cons	3117.191	40.976	76.07	0.000	3036.88	3197.503

(*Continued on next page*)

```
Variance at level 1
------------------------------------------------------------------------------
  137392.81 (2867.2533)

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


***level 2 (momid)

    var(1): 114763.41 (4266.0686)
------------------------------------------------------------------------------
```

To compare the variance estimates with the standard deviation estimates from `xtmixed` on page 134 of the book, we square the latter and display them in the same order as they appear in the `gllamm` output:

```
. display 370.6656^2
137392.99

. display 338.7669^2
114763.01
```

All estimates are close to those from `xtreg` and `xtmixed`. Estimation takes a relatively long time for this example, so if there is any chance we may need the estimates again—for instance to perform diagnostics—we should keep the estimates in memory for later use:

```
. estimates store gllamm
```

If estimates may be required in a future Stata session, they can also be saved in a file using `estimates save`.

3.6 Hypothesis tests and confidence intervals

Almost all standard post-estimation commands (e.g., `lrtest`, `lincom`, `test`) work with `gllamm`, and this is shown for `testparm` in section 3.6.1. However, the `margins` and `marginsplot` commands used in section 3.6.2 of the book do not work with `gllamm`.

3.6.1 Hypothesis tests for regression coefficients

Joint hypothesis tests for several regression coefficients

The `testparm` command works after estimation using `gllamm` in the same way as it does after estimation using `xtmixed`:

```
. testparm kessner2 kessner3

 ( 1)  [birwt]kessner2 = 0
 ( 2)  [birwt]kessner3 = 0

           chi2(  2) =   26.94
         Prob > chi2 =    0.0000
```

3.9 Assigning values to random effects: Residual diagnostics

We begin by retrieving the gllamm estimates stored in section 3.4.3 (this is not necessary here since we have not estimated any other models since estimating the model for which we want predictions):

```
. estimates restore gllamm
(results gllamm are active now)
```

We then use the gllapred command with the ustd option to obtain $r_j^{(2)}$:

```
. gllapred lev2, ustd
(means and standard deviations will be stored in lev2m1 lev2s1)
Non-adaptive log-likelihood: -65145.743
-6.515e+04 -6.515e+04
log-likelihood:-65145.752
```

and gllapred with the pearson option to obtain $r_{ij}^{(1)}$:

```
. gllapred lev1, pearson
(residuals will be stored in lev1)
Non-adaptive log-likelihood: -65145.743
-6.515e+04 -6.515e+04
log-likelihood:-65145.752
```

Histograms of the standardized level-1 residuals $r_{ij}^{(1)}$ and the standardized level-2 residuals $r_j^{(2)}$ can be plotted as follows:

```
. histogram lev1, normal xtitle(Standardized level-1 residuals)
. histogram lev2m1 if idx==1, normal xtitle(Standardized level-2 residuals)
```

These commands produce figures 3.6 and 3.7 in the book.

Chapter 4

Random-coefficient models

4.5 (g) Estimation using gllamm

4.5.1 (g) Random-intercept model

We start by using `gllamm` to fit the random-intercept model:

```
. gllamm gcse lrt, i(school) adapt
number of level 1 units = 4059
number of level 2 units = 65

Condition Number = 35.786606

gllamm model

log likelihood = -14024.799
```

gcse	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
lrt	.5633697	.0124863	45.12	0.000	.538897	.5878425
_cons	.0239115	.4002945	0.06	0.952	-.7606514	.8084744

```
Variance at level 1
------------------------------------------------------------------------------
  56.572669 (1.2662546)

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

***level 2 (school)

    var(1): 9.2127071 (1.8529779)
------------------------------------------------------------------------------
```

We store the estimates under the name `RI`:

```
. estimates store RI
```

For comparison, the variance estimates from `xtmixed` are:

```
. display 7.521481^2
56.572676
. display 3.035269^2
9.2128579
```

All estimates are practically identical to those from `xtmixed` (see page 194 of the book).

4.5.2 (g) Random-coefficient model

In the previous `gllamm` command for the random-intercept model, all that was required to specify the random intercept was the `i(school)` option.

To introduce a random slope ζ_{2j}, we will also need to specify the variable multiplying the random slope in equation (4.1) on page 188 of the book, i.e., x_{ij} or `lrt`. This is done by specifying an equation for the slope:

```
. eq slope: lrt
```

We also need an equation for the variable multiplying the random intercept ζ_{1j}, and since it is an intercept, we just specify a variable equal to 1:

```
. generate cons = 1
. eq inter: cons
```

We must also add a new option, `nrf(2)`, which stands for "number of random effects is 2" (an intercept and a slope), and specify both equations, `inter` and `slope`, in the `eqs()` option of the `gllamm` command. Finally, with two random effects, 8-point quadrature requires $8^2 = 64$ terms to evaluate the log likelihood. We can get nearly the same accuracy with only 44 points (taking about 30% less time to run) by using a spherical quadrature rule of degree 15 (see Rabe-Hesketh and Skrondal 2012b, sec. 10.11), specified using the options `ip(m)` and `nip(15)`:

```
. gllamm gcse lrt, i(school) nrf(2) eqs(inter slope) ip(m) nip(15)
>   adapt

number of level 1 units = 4059
number of level 2 units = 65

Condition Number = 35.440557

gllamm model

log likelihood = -14004.613
```

gcse	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
lrt	.556729	.0199969	27.84	0.000	.5175357	.5959223
_cons	-.1150847	.3982901	-0.29	0.773	-.8957188	.6655495

(*Continued on next page*)

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

  55.365323 (1.2492818)

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


***level 2 (school)

    var(1): 9.0447094 (1.8310164)
    cov(2,1): .18040343 (.06915233) cor(2,1): .49754032

    var(2): .01453578 (.00457732)
------------------------------------------------------------------------------
```

The `gllamm` output gives the maximum likelihood estimates of the within-school variance θ under `Variance at level 1` and the estimates of the elements ψ_{11}, ψ_{21}, and ψ_{22}, of the covariance matrix of the random intercept and slope under `Variances and covariances of random effects`. The estimated correlation between the random intercept and slope is also give as `cor(2,1): .49754323`. This correlation and the regression coefficients agree well with the estimates from `xtmixed` on page 196 of the book.

To compare the remaining estimates, we display the variance estimates from `xtmixed` in the order in which they appear in the `gllamm` output:

```
. display 7.440787^2
55.365311
. display 3.007444^2
9.0447194
. display .1205646^2
.01453582
```

These estimates are practically identical to those from `gllamm` (see output from `estat recovariance` on page 197 of the book).

We store the estimates for later use:

```
. estimates store RC
```

To speed up estimation, we could also have used the previous estimates for the random-intercept model as starting values for the regression coefficients and random-intercept variance, and set the starting values for the additional two parameters (for the random-slope variance and the random intercept and slope covariance) to zero:

```
. matrix a = e(b)
. matrix a = (a,0,0)
```

(The order in which the parameters are given in the matrix matters, and when going from a random-intercept to a random-coefficient model, the two new parameters are always at the end.) To use the parameter matrix `a` as starting values, we would specify the `from(a)` and `copy` options. The `gllamm` command would be:

```
gllamm gcse lrt, i(school) nrf(2) eqs(inter slope) ip(m) nip(15)
   adapt from(a) copy
```

4.6 Testing the slope variance

We can use a likelihood-ratio test to test the null hypothesis that the random slope variance is zero:

```
. lrtest RI RC
Likelihood-ratio test                                 LR chi2(2)  =     40.37
(Assumption: RI nested in RC)                         Prob > chi2 =    0.0000
```

The correct asymptotic p-value is obtained using

```
. display 0.5*chi2tail(1,40.37) + 0.5*chi2tail(2,40.37)
9.616e-10
```

4.8 Assigning values to the random intercepts and slopes

4.8.2 Empirical Bayes prediction

To obtain predictions based on the random-coefficient model, we first retrieve the estimates:

```
. estimates restore RC
(results RC are active now)
```

and then use the `gllapred` command with the `u` option. We specify the `stub` (or prefix) of the variables that will be generated by `gllapred`. Because the model contains two random effects, a random intercept and a random slope, four variables will be generated, two for the posterior means and two for the posterior standard deviations.

```
. gllapred eb, u
(means and standard deviations will be stored in ebm1 ebs1 ebm2 ebs2)
Non-adaptive log-likelihood: -14007.032
-1.401e+04 -1.400e+04 -1.400e+04 -1.400e+04
log-likelihood:-14004.613
```

The output tells us the variable names for the predictions. The order of the random effects is determined by the order in which they are listed in the `eqs()` option in `gllamm`. Here, the first random effect is the random intercept and the second is the random slope (note that the random intercept always comes last in `xtmixed`). We display the corresponding posterior means (as on page 202 of the book) using:

(*Continued on next page*)

```
. egen pickone = tag(school)
. list school ebm1 ebm2 if pickone==1 & (school<10 | school==48), noobs
```

school	ebm1	ebm2
1	3.749336	.12497567
2	4.7021287	.16472645
3	4.7976842	.08086626
4	.35024886	.12718282
5	2.4628061	.07205773
6	5.1838152	.05862374
7	3.640946	-.1488712
8	-.1218854	.00688557
9	-1.7679824	-.08861967
48	-.40981905	-.00648538

4.8.3 Model visualization

We can use `gllapred` with the `linpred` option to predict school-specific regression lines by plugging parameter estimates and empirical Bayes predictions of the random intercepts and slopes into the model:

```
. gllapred muRC, linpred
(linear predictor will be stored in muRC)
Non-adaptive log-likelihood: -14007.032
-1.401e+04 -1.400e+04 -1.400e+04 -1.400e+04
log-likelihood:-14004.613
```

The following command produces the right panel of figure 4.10 on page 204 of the book:

```
. sort school lrt
. twoway (line muRC lrt, connect(ascending)), xtitle(LRT)
>   ytitle(Empirical Bayes regression lines for model 2)
```

To produce the left panel (for the random-intercept model), use:

```
. estimates restore RI
(results RI are active now)
. gllapred muRI, linpred
(linear predictor will be stored in muRI)
Non-adaptive log-likelihood: -14028.596
-1.404e+04 -1.403e+04 -1.403e+04 -1.403e+04 -1.403e+04 -1.403e+04
-1.403e+04 -1.402e+04 -1.402e+04 -1.402e+04
log-likelihood:-14024.799
. sort school lrt
. twoway (line muRI lrt, connect(ascending)), xtitle(LRT)
>   ytitle(Empirical Bayes regression lines for model 1)
```

Chapter 6

Marginal models

Most of the models considered in this chapter cannot be estimated using `gllamm` because they relax the assumption that the level-1 residuals are uncorrelated which is not possible in `gllamm`. However, `gllamm` can estimate models with random-intercept or random-coefficient structures as shown in chapter 4, and it does allow the residual variance to depend on covariates, as shown in sections 6.4.2 and 6.4.3 below.

6.4 Hybrid and complex marginal models

6.4.2 Heteroscedastic level-1 residuals over occasions

In `gllamm`, we can specify a model for the logarithm of the level-1 standard deviation of the form

$$\ln\sqrt{\theta_i} = \frac{1}{2}\ln\theta_i \ = \ \alpha_1 x_{1i} + \alpha_2 x_{2i} + \cdots$$

If the variables x_{1i}, etc., are dummy variables, then we get a different standard deviation for each of the corresponding groups. Note that the equation does not contain an intercept, so we can use dummy variables for all groups and do not have to omit one dummy variable as usual.

To obtain different standard deviations for different years, we first generate dummy variables for years:

```
. tabulate yeart, generate(yr)
```

yeart	Freq.	Percent	Cum.
0	545	12.50	12.50
1	545	12.50	25.00
2	545	12.50	37.50
3	545	12.50	50.00
4	545	12.50	62.50
5	545	12.50	75.00
6	545	12.50	87.50
7	545	12.50	100.00
Total	4,360	100.00	

Now we can define an equation, called **het**, for the log-standard deviation by specifying the eight dummy variables for the years:

```
. eq het: yr1-yr8
```

This equation will be passed to `gllamm` using the `s()` option.

To specify a random-coefficient model, we also have to define equations for the intercept and slope to pass to `gllamm` using the `eqs()` option:

```
. generate one = 1
. eq inter: one
. eq slope: yeart
```

We use degree 11 spherical quadrature (`ip(m)` and `nip(11)` options). This takes some time to run, and you could also try `nip(7)` or even `nip(5)` to speed estimation (often `nip(5)` is not sufficiently accurate, but here it works quite well). The `gllamm` command is:

```
. gllamm lwage black hisp union married exper yeart educt, i(nr)
>     nrf(2) eqs(inter slope) s(het) ip(m) nip(11) adapt


number of level 1 units = 4360
number of level 2 units = 545

Condition Number = 36.573837

gllamm model

log likelihood = -2036.413
```

lwage	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
black	-.1409611	.0485077	-2.91	0.004	-.2360344	-.0458878
hisp	.0197168	.0432223	0.46	0.648	-.0649974	.1044311
union	.1033512	.0175008	5.91	0.000	.0690503	.1376521
married	.0749567	.0168062	4.46	0.000	.0420171	.1078963
exper	.031004	.011544	2.69	0.007	.0083782	.0536299
yeart	.0264718	.0119172	2.22	0.026	.0031144	.0498291
educt	.0945191	.0108146	8.74	0.000	.0733228	.1157154
_cons	1.333682	.0393056	33.93	0.000	1.256645	1.41072

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

    equation for log standard deviation:

    yr1: -.78340246 (.03602338)
    yr2: -1.0332121 (.03698541)
    yr3: -1.2444805 (.03797054)
    yr4: -1.2750318 (.0360111)
    yr5: -1.0983855 (.03371716)
    yr6: -1.1747076 (.03487452)
    yr7: -1.1350565 (.03609228)
    yr8: -1.4489631 (.05324738)
```

(*Continued on next page*)

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


***level 2 (nr)

    var(1): .14474557 (.01229625)
    cov(2,1): -.00966248 (.001685) cor(2,1): -.47235711

    var(2): .00289089 (.00033276)
------------------------------------------------------------------------------
```

To compare the level-2 variances with the estimated standard deviations on page 318 of the book from `xtmixed`, we square the latter:

```
. display .3804544^2
.14474555
. display .053767^2
.00289089
```

To see what the estimated level-1 standard deviations are (instead of their logarithms), we can use:

```
. display exp([lns1]yr1)
.45684895
. display exp([lns1]yr2)
.35586207
. display exp([lns1]yr3)
.28809053
. display exp([lns1]yr4)
.27942208
. display exp([lns1]yr5)
.33340892
. display exp([lns1]yr6)
.30890929
. display exp([lns1]yr7)
.32140397
. display exp([lns1]yr8)
.23481364
```

All estimates are practically identical to those from `xtmixed`.

6.4.3 Heteroscedastic level-1 residuals over groups

Similarly, for the random-intercept model with different residual variances for the three ethnicities, use:

```
. generate white = 1 - black - hisp
. eq het: white black hisp
```

(*Continued on next page*)

```
. gllamm lwage black hisp union married exper yeart educt, i(nr)
>     s(het) adapt

number of level 1 units = 4360
number of level 2 units = 545

Condition Number = 39.649215

gllamm model

log likelihood = -2213.0207
```

lwage	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
black	-.133563	.0480979	-2.78	0.005	-.2278331	-.039293
hisp	.0175076	.042645	0.41	0.681	-.066075	.1010902
union	.1083187	.0179863	6.02	0.000	.0730662	.1435712
married	.0751363	.0167458	4.49	0.000	.0423152	.1079574
exper	.0331674	.0111967	2.96	0.003	.0112222	.0551126
yeart	.0260171	.011402	2.28	0.023	.0036695	.0483646
educt	.0946593	.0107039	8.84	0.000	.0736801	.1156385
_cons	1.317362	.0373883	35.23	0.000	1.244082	1.390641

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

    equation for log standard deviation:

    white: -1.0359907 (.01342136)
    black: -1.0122484 (.0337608)
    hisp: -1.0804347 (.02895928)

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

***level 2 (nr)

    var(1): .10698415 (.00747079)
------------------------------------------------------------------------------
```

For comparison with the `xtmixed` estimates on page 319 of the book, the random-intercept variance estimate from `xtmixed` is:

```
. display .3270843^2
.10698414
```

and the level-1 standard deviation estimates from `gllamm` are:

```
. display exp([lns1]white)
.35487462

. display exp([lns1]black)
.36340098

. display exp([lns1]hisp)
.33944793
```

Again, agreement is nearly perfect.

Chapter 7

Growth-curve models

7.3 Models for nonlinear growth

7.3.1 Polynomial models

(g) Fitting the models

As usual, for random-coefficient models in `gllamm`, we first define equations for the intercept and slope and then pass them to `gllamm` using the `eqs()` option. We use degree 15 spherical adaptive quadrature:

```
. generate age2 = age^2
. generate cons = 1
. eq inter: cons
. eq slope: age
. gllamm weight girl age age2, i(id) nrf(2) eqs(inter slope)
> ip(m) nip(15) adapt

number of level 1 units = 198
number of level 2 units = 68

Condition Number = 9.151205

gllamm model

log likelihood = -253.86692
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
girl	-.5960093	.2007172	-2.97	0.003	-.9894077	-.2026108
age	7.697967	.2391093	32.19	0.000	7.229322	8.166613
age2	-1.657843	.0885717	-18.72	0.000	-1.83144	-1.484245
_cons	3.794769	.1672318	22.69	0.000	3.467001	4.122537

(*Continued on next page*)

```
Variance at level 1
------------------------------------------------------------------------------
  .32756175 (.05680649)

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


***level 2 (id)

    var(1): .35370507 (.15342767)
    cov(2,1): .04762586 (.08751458) cor(2,1): .15710858

    var(2): .25980321 (.08887197)
------------------------------------------------------------------------------
. estimates store RC
```

The log likelihood and estimates of the regression coefficients are practically identical to those on page 347 of the book from `xtmixed`, as is the estimated correlation between random intercept and slope. We can obtain the estimated level-1 variance, random-intercept and random-slope variance from `xtmixed`, respectively, using:

```
. display .57233^2
.32756163
. display .5947313^2
.35370532
. display .5097091^2
.25980337
```

and these are all very close to the `gllamm` estimates.

(g) Predicting trajectories for individual children

We can use `gllapred` with the `linpred` option to predict the individual growth trajectories:

```
. gllapred traj, linpred
(linear predictor will be stored in traj)
Non-adaptive log-likelihood: -253.94794
 -253.9000  -253.8669  -253.8669
log-likelihood:-253.86692
```

Figure 7.5 on page 352 of the book is produced as follows:

```
. twoway (scatter weight age) (line traj age, sort) if girl==1,
>  by(id, compact legend(off))
```

7.4 Heteroscedasticity

7.4.1 Heteroscedasticity at level 1

As shown in section 6.4.2 of this document, we can use the `s()` option in `gllamm` to specify a linear model for the logarithm of the level-1 standard deviation. The model is defined using the `eq` command. No intercept is included, so to specify different log standard deviations for boys and girls, we use a dummy variable for each gender (we do not omit one dummy variable):

```
. generate boy = 1 - girl
. eq het: boy girl
. eq inter: cons
. eq slope: age
. gllamm weight girl age age2, i(id) nrf(2) eqs(inter slope)
>    s(het) ip(m) nip(15) adapt

number of level 1 units = 198
number of level 2 units = 68

Condition Number = 9.6099371

gllamm model

log likelihood = -252.40553
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
girl	-.6026741	.2049726	-2.94	0.003	-1.004413	-.2009353
age	7.629066	.2368424	32.21	0.000	7.164863	8.093268
age2	-1.635112	.0874551	-18.70	0.000	-1.806521	-1.463703
_cons	3.829123	.1757713	21.78	0.000	3.484617	4.173628

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

    equation for log standard deviation:

    boy: -.4467279 (.10906431)
    girl: -.70562715 (.11784509)

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


***level 2 (id)

    var(1): .39329041 (.14805673)
    cov(2,1): .04438919 (.08252851) cor(2,1): .14571939

    var(2): .23594266 (.08638896)
------------------------------------------------------------------------------
```

We can display the estimated residual standard deviations for boys and girls by exponentiating the estimated log-standard deviations:

```
. display exp([lns1]boy)
.63971795

. display exp([lns1]girl)
.49379879
```

These estimates are practically identical to those from `xtmixed` on page 361 of the book as are the level-2 variance estimates, given below for `xtmixed`:

```
. display .6271286^2
.39329028

. display .4857393^2
.23594267
```

The likelihood-ratio test, comparing this model with the previous random-coefficient model (with homoscedastic level-1 residuals) is obtained using:

```
. lrtest RC .
Likelihood-ratio test                                 LR chi2(1)  =      2.92
(Assumption: RC nested in .)                          Prob > chi2 =    0.0873
```

7.4.2 Heteroscedasticity at level 2

For heteroscedasticity at level 2, we can use the same trick in `gllamm` as we used in `xtmixed`. We specify two random parts, one for boys and one for girls. In each random part, the random intercept is multiplied by the dummy variable for the corresponding gender and the random slope is multiplied by age times the dummy variable for the corresponding gender. As in `xtmixed`, we can specify two random parts for the same level by pretending that we have a three-level model, but specifying the same cluster identifier for levels 2 and 3.

We first create new variables `age_boy` and `age_girl` for the random slopes for boys and girls, respectively, and then define equations for the random intercept and slope for boys (`intboy` and `slpby`) and for the random intercept and slope for girls (`intgirl` and `slpgirl`):

```
. generate age_boy = age*boy
. generate age_girl = age*girl
. eq intboy: boy
. eq slboy: age_boy
. eq intgirl: girl
. eq slgirl: age_girl
```

In the `gllamm` command we use the `i(id id)` option to specify two levels of nesting (levels 2 and 3) both with the same identifier `id`. The `nrf()` option then expects two numbers, `nrf(# #)`, for the number of random effects at levels 2 and 3. Here we have a random intercept and slope at each level so we use `nrf(2 2)`. We will use "level 2" for boys and "level 3" for girls, so our `eqs()` option will list the equation names in this order: `eqs(intboy slboy intgirl slgirl)`.

The model has a total of four random effects which can take a long time to estimate in `gllamm`. We therefore specify spherical quadrature of degree 5 (which happens to give good estimates for this dataset):

```
. gllamm weight girl age age2, i(id id) nrf(2 2)
>    eqs(intboy slboy intgirl slgirl)
>    ip(m) nip(5) adapt

number of level 1 units = 198
number of level 2 units = 68
number of level 3 units = 68

Condition Number = 11.906761
```

(*Continued on next page*)

```
gllamm model

log likelihood = -249.70705
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
girl	-.6067896	.2038118	-2.98	0.003	-1.006253	-.2073258
age	7.613138	.2364782	32.19	0.000	7.14965	8.076627
age2	-1.646243	.0874661	-18.82	0.000	-1.817673	-1.474812
_cons	3.820108	.162376	23.53	0.000	3.501857	4.138359

```
Variance at level 1
------------------------------------------------------------------------------
  .32178303 (.05248488)

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


***level 2 (id)

    var(1): .28830336 (.18984704)
    cov(2,1): .01825417 (.13419043) cor(2,1): .0493191

    var(2): .47516482 (.1795417)

***level 3 (id)

    var(1): .48912345 (.22086607)
    cov(2,1): .05868205 (.08335526) cor(2,1): .38227205

    var(2): .0481778 (.054581)
------------------------------------------------------------------------------
```

The estimates are remarkably close to those from `xtmixed` on page 363 of the book, given that we only used a degree 5 integration rule. The estimates from `xtmixed` of the five variances, in the order they are given in the output above are:

```
. display .567544^2
.32210619

. display .5364599^2
.28778922

. display .6891223^2
.47488954

. display .6987662^2
.4882742

. display .218154^2
.04759117
```

The likelihood-ratio test comparing this model with the original random-coefficient model can be obtained as follows:

```
. lrtest RC .

Likelihood-ratio test                                 LR chi2(3)  =      8.32
(Assumption: RC nested in .)                          Prob > chi2 =    0.0398
```

❖ Alternative model for heteroscedasticity at level 2

We can also fit a model that allows the random-intercept and random-slope variances to differ between the genders but not the correlation between random intercept and random slope. This can be accomplished by using *factor loadings* in `gllamm`.

The random part of the model can be written as

$$\zeta_{1j}(\texttt{boy}_j + \lambda_1\texttt{girl}_j) + \zeta_{2j}(t_{ij}\texttt{boy}_j + \lambda_2 t_{ij}\texttt{girl}_j) + \epsilon_{ij} \;=\; \begin{cases} \zeta_{1j} + \zeta_{2j}t_{ij} + \epsilon_{ij} & \text{if } \texttt{boy}_j = 1 \\ \zeta_{1j}\lambda_1 + \zeta_{2j}\lambda_2 t_{ij} + \epsilon_{ij} & \text{if } \texttt{girl}_j = 1 \end{cases}$$

where λ_1 and λ_2 are factor loadings. Note that linear mixed models do not allow for factor loadings, so this is no longer a standard linear mixed model.

In `gllamm` we specify the terms multiplying the random effects through the `eq` command. For the random intercept, we specify the variables `boy` and `girl`:

```
. eq inter: boy girl
```

`gllamm` automatically sets the factor loading for the first variable to one.

For the random slope, we specify the variables `age_boy` and `age_girl`:

```
. eq slope: age_boy age_girl
```

The model now includes only two random effects and is faster to estimate. We can therefore use degree 15 spherical integration:

```
. gllamm weight girl age age2, i(id) nrf(2) eqs(inter slope)
> ip(m) nip(15) adapt

number of level 1 units = 198
number of level 2 units = 68

Condition Number = 19.88205

gllamm model

log likelihood = -249.80965
```

weight	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
girl	-.6093703	.2058006	-2.96	0.003	-1.012732	-.2060086
age	7.623536	.2358429	32.32	0.000	7.161292	8.085779
age2	-1.648115	.0872424	-18.89	0.000	-1.819107	-1.477123
_cons	3.809033	.1592697	23.92	0.000	3.49687	4.121196

```
Variance at level 1
------------------------------------------------------------------------------
  .3202304 (.05462852)

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


***level 2 (id)

    var(1): .26625045 (.18083572)
```

```
    loadings for random effect 1
    boy: 1 (fixed)
    girl: 1.4211837 (.48836587)

    cov(2,1): .04948418 (.11335276) cor(2,1): .14211781

    var(2): .45535006 (.17310296)

    loadings for random effect 2
    age_boy: 1 (fixed)
    age_girl: .3644414 (.17706435)

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

The factor loadings are estimated as $\widehat{\lambda}_1 = 1.4211837$ and $\widehat{\lambda}_1 = 0.3644414$. We see from the log likelihood that the model fits nearly as well as the model that allowed the correlation between random intercept and random slope to differ between boys and girls in addition to the variances.

We can obtain the estimated random-intercept variance for girls using:

```
. display .26625045*1.4211837^2
.53776284
```

and the estimated random slope variance for girls using:

```
. display .45535006*.3644414^2
.06047847
```

The likelihood-ratio test comparing this model with the original random-coefficient model can be obtained as follows:

```
. lrtest RC .

Likelihood-ratio test                                 LR chi2(2)  =      8.11
(Assumption: RC nested in .)                          Prob > chi2 =    0.0173
```

Chapter 8

Higher-level models with nested random effects

8.6 (g) Estimation using gllamm

In `gllamm`, the levels of the model are defined by listing the cluster identifiers in the `i()` option, starting with level 2, then level 3, etc. Note that the levels are specified in the opposite order in `xtmixed`. By default, the model has a random intercept at each level. The syntax to fit the three-level variance components model to the peak expiratory flow data is:

```
. gllamm w, i(method id) adapt

number of level 1 units = 68
number of level 2 units = 34
number of level 3 units = 17

Condition Number = 224.35613

gllamm model

log likelihood = -345.29005
```

w	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
_cons	450.8971	26.6384	16.93	0.000	398.6867	503.1074

(*Continued on next page*)

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

  315.36764 (76.487891)

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


***level 2 (method)

    var(1): 379.32392 (188.12061)

***level 3 (id)

    var(1): 11794.773 (4138.6952)
------------------------------------------------------------------------------
```

The estimates are practically identical to those from `xtmixed` on page 393 of the book. To compare the variance estimates, we square the standard deviation estimates from `xtmixed`, in the order that they are displayed in `gllamm`:

```
. display 17.75859^2
315.36752
. display 19.47623^2
379.32354
. display 108.6037^2
11794.764
```

8.7 Empirical Bayes prediction

The `gllapred` command with the `u` option produces empirical Bayes predictions of the random effects in the order that they are specified in the `gllamm` command:

```
. gllapred reff, u
(means and standard deviations will be stored in reffm1 reffs1 reffm2 reffs2)
Non-adaptive log-likelihood: -351.36155
 -444.5225  -424.4041  -406.6573  -390.3798  -377.2486  -366.6731
 -358.0680  -351.7793  -346.8303  -345.2988  -345.2901  -345.2901
log-likelihood:-345.29005
```

The variable `reffm1` contains the posterior means (empirical Bayes predictions) of the method-level random intercepts and the variable `reffm2` contains the posterior means of the subject-level random intercepts. We can list these predictions as on page 395 of the book using:

(*Continued on next page*)

```
. sort id method
. label define m 0 "Wright" 1 "Mini Wright"
. label values method m
. list id method reffm2 reffm1 if id<8 & occasion==1, noobs sepby(id)
```

id	method	reffm2	reffm1
1	Wright	53.143156	-8.5047963
1	Mini Wright	53.143156	10.213898
2	Wright	-40.72008	-10.014133
2	Mini Wright	-40.72008	8.7045614
3	Wright	61.698399	.99212078
3	Mini Wright	61.698399	.99212078
4	Wright	-23.609595	-6.9135302
4	Mini Wright	-23.609595	6.1542376
5	Wright	34.810493	-8.9761801
5	Mini Wright	34.810493	10.095697
6	Wright	144.07317	-7.748992
6	Mini Wright	144.07317	12.382434
7	Wright	-37.053548	.11053846
7	Mini Wright	-37.053548	-1.3021932

8.8 Testing variance components

We can use a likelihood-ratio test for the null hypothesis that the variance for methods within subjects is zero:

```
. estimates store THR
. quietly gllamm w, i(id) adapt
. lrtest THR .
Likelihood-ratio test                                 LR chi2(1)  =      9.20
(Assumption: . nested in THR)                         Prob > chi2 =    0.0024
```

8.9 Crossed versus nested random effects revisited

The `gllamm` command for fitting the model with a fixed effect for `method` is:

```
. gllamm w method, i(method id) adapt

number of level 1 units = 68
number of level 2 units = 34
number of level 3 units = 17

Condition Number = 227.30894
```

(*Continued on next page*)

```
gllamm model

log likelihood = -344.99736
```

w	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
method	6.029412	7.812741	0.77	0.440	-9.28328	21.3421
_cons	447.8824	26.9233	16.64	0.000	395.1136	500.6511

```
Variance at level 1
------------------------------------------------------------------------------
  315.36765 (76.487893)

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

***level 2 (method)

    var(1): 361.14706 (182.02062)

***level 3 (id)

    var(1): 11803.861 (4138.627)
------------------------------------------------------------------------------
```

To compare the variance estimates with those from `xtmixed` on page 398 of the book, we display the latter in the same order as they appear in the `gllamm` output:

```
. display 17.75859^2
315.36752
. display 19.00386^2
361.14669
. display 108.6455^2
11803.845
```

Again, the estimates are very close.

8.12 Three-level random-intercept model

8.12.3 (g) Estimation using gllamm

When the model contains only random intercepts, specification of the random part of the model requires only the `i()` option, here `i(id schoolid)`. We use 5 quadrature points per dimension to speed estimation:

```
. gllamm ravens meat milk calorie relyear meat_year milk_year calorie_year
>    age_at_time0 boy, i(id schoolid) nip(5) adapt

number of level 1 units = 2593
number of level 2 units = 542
number of level 3 units = 12

Condition Number = 43.636436
```

(*Continued on next page*)

```
gllamm model

log likelihood = -6255.8919

------------------------------------------------------------------------------
      ravens |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        meat |  -.2557746   .3331808    -0.77   0.443     -.908797    .3972478
        milk |  -.4792254   .3224457    -1.49   0.137    -1.111207    .1527565
     calorie |   -.371442   .3293026    -1.13   0.259    -1.016863    .2739792
     relyear |   .9131073   .1406552     6.49   0.000     .6374281    1.188786
   meat_year |   .5111623   .1967369     2.60   0.009     .1255651    .8967595
   milk_year |  -.1057321   .1914634    -0.55   0.581    -.4809934    .2695293
calorie_year |   .1226815   .1931934     0.64   0.525    -.2559706    .5013337
 age_at_time0 |  .1163292   .0616601     1.89   0.059    -.0045223    .2371808
         boy |   .5372866    .165845     3.24   0.001     .2122363    .8623369
       _cons |   16.70636    .509331    32.80   0.000     15.70809    17.70463
------------------------------------------------------------------------------

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

  5.8223567 (.1816606)

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


***level 2 (id)

    var(1): 2.3472399 (.22325898)

***level 3 (schoolid)

    var(1): .04883539 (.05906999)
------------------------------------------------------------------------------

. estimates store M1
```

In the order that they appear above, the variance estimates from `xtmixed` on page 407 of the book are:

```
. display 2.412956^2
5.8223567
. display 1.532073^2
2.3472477
. display .2209702^2
.04882783
```

8.13 Three-level random-coefficient models

8.13.1 Random coefficient at the child-level

We now add a random slope for `relyear` at the level 2. We therefore have to define an equation for the random intercept and an equation for the random slope:

```
. generate cons = 1
. eq inter: cons
```

```
. eq slope: relyear
```

In the `gllamm` command we also need the `nrf()` option to specify how many random effects there are at each level, here `nrf(2 1)`, and the `eqs()` option to list the corresponding three equations, here `eqs(inter slope inter)`:

```
. gllamm ravens meat milk calorie relyear meat_year milk_year calorie_year
>    age_at_time0 boy, i(id schoolid) nrf(2 1) eqs(inter slope inter)
>    ip(m) nip(5) adapt

number of level 1 units = 2593
number of level 2 units = 542
number of level 3 units = 12

Condition Number = 41.753006

gllamm model

log likelihood = -6241.3374
```

ravens	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
meat	-.236802	.3401249	-0.70	0.486	-.9034345	.4298305
milk	-.4783402	.3304776	-1.45	0.148	-1.126064	.1693841
calorie	-.3685747	.3382833	-1.09	0.276	-1.031598	.2944484
relyear	.9236107	.1567281	5.89	0.000	.6164293	1.230792
meat_year	.4990581	.2194573	2.27	0.023	.0689296	.9291865
milk_year	-.1156097	.2139669	-0.54	0.589	-.5349772	.3037578
calorie_year	.1169182	.2149197	0.54	0.586	-.3043168	.5381531
age_at_time0	.1146557	.0609246	1.88	0.060	-.0047544	.2340658
boy	.4902233	.1648488	2.97	0.003	.1671255	.813321
_cons	16.74273	.5069814	33.02	0.000	15.74906	17.73639

```
Variance at level 1
------------------------------------------------------------------------------
  5.3609699 (.19291996)

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

***level 2 (id)

    var(1): 2.1263441 (.27162165)
    cov(2,1): -.00139043 (.17591362) cor(2,1): -.00110335

    var(2): .74685559 (.19683476)

***level 3 (schoolid)

    var(1): .06557091 (.0717574)
------------------------------------------------------------------------------
. estimates store M2
```

The corresponding variance estimates from `xtmixed` on page 410 of the book are:

```
. display 2.315432^2
5.3612253
. display 1.45822^2
2.1264056
. display .864055^2
```

```
.74659104

. display .2546548^2
.06484907
```

The likelihood-ratio test for comparing the three-level random-intercept model with the three-level model that has a random slope at the child level can be performed using:

```
. lrtest M1 M2

Likelihood-ratio test                                 LR chi2(2)  =     29.11
(Assumption: M1 nested in M2)                         Prob > chi2 =    0.0000
```

8.13.2 Random coefficient at the child and school levels

We now add a random slope of `relyear` at the school level. The resulting model has four random effects and estimation is slow, so we use spherical quadrature of degree 5. We also use the estimates from the previous model as starting values. We can do that very easily here because the previous model used the same equation names for the random effects that the models share in common, so the parameter vector will have the correct equation names and column names. The starting values for the new parameters (for variance of the new random slope and the covariance between the school-level random intercept and random slope) will be set to zero. The commands are:

```
. matrix a=e(b)

. gllamm ravens meat milk calorie relyear meat_year milk_year calorie_year
>    age_at_time0 boy, i(id schoolid) nrf(2 2) eqs(inter slope inter slope)
>    ip(m) nip(5) from(a) adapt

number of level 1 units = 2593
number of level 2 units = 542
number of level 3 units = 12

Condition Number = 49.123752

gllamm model

log likelihood = -6240.6132
```

ravens	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
meat	-.226835	.3783363	-0.60	0.549	-.9683604	.5146905
milk	-.4940559	.3623273	-1.36	0.173	-1.204204	.2160925
calorie	-.3500534	.3805169	-0.92	0.358	-1.095853	.3957461
relyear	.9211458	.1698552	5.42	0.000	.5882357	1.254056
meat_year	.4751492	.2391628	1.99	0.047	.0063987	.9438997
milk_year	-.1087991	.2320662	-0.47	0.639	-.5636406	.3460423
calorie_year	.1043165	.2361275	0.44	0.659	-.3584848	.5671178
age_at_time0	.1131252	.0607745	1.86	0.063	-.0059906	.2322409
boy	.4906368	.164508	2.98	0.003	.168207	.8130666
_cons	16.76484	.5194477	32.27	0.000	15.74674	17.78294

(*Continued on next page*)

```
Variance at level 1
------------------------------------------------------------------------------
  5.3603882 (.1929498)

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


***level 2 (id)

    var(1): 2.100472 (.27199282)
    cov(2,1): .01673053 (.17561439) cor(2,1): .01346052

    var(2): .73549376 (.19663458)

***level 3 (schoolid)

    var(1): .12262135 (.15586286)
    cov(2,1): -.04258019 (.06022169) cor(2,1): -.99996871

    var(2): .01478687 (.02732136)
------------------------------------------------------------------------------
```

The estimates differ a little from those produced by `xtmixed` (see page 412 of the book). We can get estimates that are very close to the `xtmixed` estimates by using `nip(7)` instead of `nip(5)` but this will take longer to run.

The variance estimates from `xtmixed` are:

```
. display 2.31526^2
5.3604289

. display 1.452461^2
2.109643

. display .8583323^2
.73673434

. display .3172716^2
.10066127

. display .1110955^2
.01234221
```

We restore the estimates of the previous model

```
. estimates restore M2
(results M2 are active now)
```

8.14 Residual diagnostics and predictions

We can obtain empirical Bayes predictions of the random effects using:

(*Continued on next page*)

```
. predict eb, u
(means and standard deviations will be stored in ebm1 ebs1 ebm2 ebs2 ebm3 ebs3)
Non-adaptive log-likelihood: -6289.1885
-6321.1680 -6308.2454 -6300.9731 -6296.6789 -6293.2117 -6290.3333
-6287.8986 -6285.8287 -6284.1725 -6282.9172 -6281.8684 -6280.7653
-6279.4592 -6277.9273 -6276.2307 -6274.4521 -6272.6437 -6270.8288
-6269.0166 -6267.2111 -6265.4146 -6263.6288 -6261.8554 -6260.0966
-6258.3548 -6256.6333 -6254.9363 -6253.2692 -6251.6395 -6250.0578
-6248.5390 -6247.1046 -6245.7862 -6244.6269 -6243.6707 -6242.9229
-6242.3293 -6241.8587 -6241.5362 -6241.3796 -6241.3408 -6241.3379
-6241.3379
log-likelihood:-6241.3379
```

To keep track of which variable is for which random effect, we rename the variables produced by `gllapred`:

```
. rename ebm1 RI2
. rename ebm2 RC2
. rename ebm3 RI3
```

For the level-1 residuals, we can only get standardized residuals by using the `pearson` option:

```
. predict RES, pearson
(residuals will be stored in RES)
Non-adaptive log-likelihood: -6289.1885
-6321.1680 -6308.2454 -6300.9731 -6296.6789 -6293.2117 -6290.3333
-6287.8986 -6285.8287 -6284.1725 -6282.9172 -6281.8684 -6280.7653
-6279.4592 -6277.9273 -6276.2307 -6274.4521 -6272.6437 -6270.8288
-6269.0166 -6267.2111 -6265.4146 -6263.6288 -6261.8554 -6260.0966
-6258.3548 -6256.6333 -6254.9363 -6253.2692 -6251.6395 -6250.0578
-6248.5390 -6247.1046 -6245.7862 -6244.6269 -6243.6707 -6242.9229
-6242.3293 -6241.8587 -6241.5362 -6241.3796 -6241.3408 -6241.3379
-6241.3379
log-likelihood:-6241.3379
```

To obtain unstandardized residuals, we multiply by the estimated residual standard deviation:

```
. replace RES = RES*exp([lns1]_cons)
(2593 real changes made)
```

We can now produce the graph in figures 8.6 of the book (page 414):

```
. replace RI3=. if pick_school!=1
(2581 real changes made, 2581 to missing)
. replace RI2 =. if rn!=1
(2059 real changes made, 2059 to missing)
. graph box RI3 RI2 RES, ascategory box(1, bstyle(outline))
>     yvaroptions(relabel(1 "School" 2 "Child" 3 "Occasion"))
>         medline(lcolor(black))
```

The graph in figure 8.7 is produced using:

```
. scatter RC2 RI2 if rn==1,  saving(yx, replace)
>    xtitle("Random intercept") ytitle("Random slope")
(file yx.gph saved)
. histogram RC2, freq horiz saving(hy, replace)
>     yscale(alt) ytitle(" ") fxsize(35) normal
(bin=34, start=-1.3361212, width=.08589759)
(file hy.gph saved)
. histogram RI2, freq saving(hx, replace)
>     xscale(alt) xtitle(" ") fysize(35) normal
(bin=23, start=-7.715339, width=.50420418)
```

```
(file hx.gph saved)
. graph combine hx.gph yx.gph hy.gph, hole(2) imargin(0 0 0 0)
```

and the graph in figure 8.8 is produced using:

```
. gllapred ptraj, linpred
(linear predictor will be stored in ptraj)
Non-adaptive log-likelihood: -6289.1885
-6321.1680 -6308.2454 -6300.9731 -6296.6789 -6293.2117 -6290.3333
-6287.8986 -6285.8287 -6284.1725 -6282.9172 -6281.8684 -6280.7653
-6279.4592 -6277.9273 -6276.2307 -6274.4521 -6272.6437 -6270.8288
-6269.0166 -6267.2111 -6265.4146 -6263.6288 -6261.8554 -6260.0966
-6258.3548 -6256.6333 -6254.9363 -6253.2692 -6251.6395 -6250.0578
-6248.5390 -6247.1046 -6245.7862 -6244.6269 -6243.6707 -6242.9229
-6242.3293 -6241.8587 -6241.5362 -6241.3796 -6241.3408 -6241.3379
-6241.3379
log-likelihood:-6241.3379
. sort schoolid id relyear
. twoway (line ptraj relyear, connect(ascending)), by(schoolid, compact)
>    xtitle(Time in years) ytitle(Raven's score)
```

References

Chung, Y., S. Rabe-Hesketh, A. Gelman, J. Liu, and V. Dorie. 2011. Avoiding boundary estimates in linear mixed models through weakly informative priors. Technical Report 284, U.C. Berkeley Division of Biostatistics Working Paper Series. Downloadable from http://www.bepress.com/ucbbiostat/paper160/.

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

Rabe-Hesketh, S., and A. Skrondal. 2006. Multilevel modeling of complex survey data. *Journal of the Royal Statistical Society, Series A* 169: 805–827.

———. 2012a. *Multilevel and Longitudinal Modeling Using Stata, Volume I.* 3rd ed. College Station, TX: Stata Press.

———. 2012b. *Multilevel and Longitudinal Modeling Using Stata, Volume II.* 3rd ed. College Station, TX: Stata Press.

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

———. 2004a. GLLAMM Manual. Technical Report 160, U.C. Berkeley Division of Biostatistics Working Paper Series. Downloadable from http://www.bepress.com/ucbbiostat/paper160/.

———. 2004b. Generalized multilevel structural equation modeling. *Psychometrika* 69: 167–190.

———. 2005. Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics* 128: 301–323.

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

———. 2007. Latent variable modelling: A survey. *Scandinavian Journal of Statistics* 34: 712–745.

———. 2009. Prediction in multilevel generalized linear models. *Journal of the Royal Statistical Society, Series A* 172: 659–687.