> **meqrpoisson** — Multilevel mixed-effects Poisson regression (QR decomposition)

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

meqrpoisson, like mepoisson, fits mixed-effects models for count responses, for which the conditional distribution of the response given the random effects is assumed to be Poisson.

meqrpoisson provides an alternative estimation method that uses the QR decomposition of the variance-components matrix. This method may aid convergence when variance components are near the boundary of the parameter space.

Quick start

Two-level Poisson regression of y on x with random intercepts by lev2 using QR decomposition
```
meqrpoisson y x || lev2:
```

As above, but report incidence-rate ratios
```
meqrpoisson y x || lev2:, irr
```

Add indicator variables for each level of categorical variable a and random coefficients on x
```
meqrpoisson y x i.a || lev2: x, irr
```

Three-level random-intercept model of y on x with lev2 nested within lev3
```
meqrpoisson y x || lev3: || lev2:
```

Menu

Statistics > Multilevel mixed-effects models > Estimation by QR decomposition > Poisson regression

Syntax

> meqrpoisson *depvar fe_equation* || *re_equation* [|| *re_equation ...*] [, *options*]

where the syntax of *fe_equation* is

> [*indepvars*] [*if*] [*in*] [, *fe_options*]

and the syntax of *re_equation* is one of the following:

> for random coefficients and intercepts
>
> > *levelvar*: [*varlist*] [, *re_options*]
>
> for random effects among the values of a factor variable
>
> > *levelvar*: R.*varname* [, *re_options*]

levelvar is a variable identifying the group structure for the random effects at that level or is _all representing one group comprising all observations.

fe_options	Description
Model	
<u>noc</u>onstant	suppress constant term from the fixed-effects equation
<u>exp</u>osure(*varname$_e$*)	include ln(*varname$_e$*) in model with coefficient constrained to 1
<u>off</u>set(*varname$_o$*)	include *varname$_o$* in model with coefficient constrained to 1

re_options	Description
Model	
<u>cov</u>ariance(*vartype*)	variance–covariance structure of the random effects
<u>noc</u>onstant	suppress constant term from the random-effects equation
<u>col</u>linear	keep collinear variables

options	Description

Reporting
<u>level</u>(#)	set confidence level; default is level(95)
irr	report fixed-effects coefficients as incidence-rate ratios
<u>var</u>iance	show random-effects parameter estimates as variances and covariances; the default
<u>stdd</u>eviations	show random-effects parameter estimates as standard deviations and correlations
<u>nore</u>table	suppress random-effects table
<u>nofe</u>table	suppress fixed-effects table
<u>estm</u>etric	show parameter estimates in the estimation metric
<u>nohe</u>ader	suppress output header
<u>nog</u>roup	suppress table summarizing groups
<u>nolr</u>test	do not perform likelihood-ratio test comparing with Poisson regression
[display_options](#)	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling

Integration
<u>intp</u>oints(# [# ...])	set the number of integration (quadrature) points; default is intpoints(7)
<u>lap</u>lace	use Laplacian approximation; equivalent to intpoints(1)

Maximization
[maximize_options](#)	control the maximization process; seldom used
<u>reto</u>lerance(#)	tolerance for random-effects estimates; default is retolerance(1e-8); seldom used
<u>reiter</u>ate(#)	maximum number of iterations for random-effects estimation; default is reiterate(50); seldom used
<u>mats</u>qrt	parameterize variance components using matrix square roots; the default
<u>matl</u>og	parameterize variance components using matrix logarithms
<u>refine</u>opts(*[maximize_options](#)*)	control the maximization process during refinement of starting values
<u>coefl</u>egend	display legend instead of statistics

vartype	Description
<u>inde</u>pendent	one unique variance parameter per random effect, all covariances 0; the default unless the R. notation is used
<u>exch</u>angeable	equal variances for random effects, and one common pairwise covariance
<u>iden</u>tity	equal variances for random effects, all covariances 0; the default if the R. notation is used
<u>un</u>structured	all variances and covariances to be distinctly estimated

indepvars may contain factor variables; see [U] **11.4.3 Factor variables**.

indepvars and *varlist* may contain time-series operators; see [U] **11.4.4 Time-series varlists**.

bootstrap, by, jackknife, mi estimate, rolling, and statsby are allowed; see [U] **11.1.10 Prefix commands**.

coeflegend does not appear in the dialog box.

See [U] **20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options

|‾ Model ‾|

noconstant suppresses the constant (intercept) term and may be specified for the fixed-effects equation and for any of or all the random-effects equations.

exposure(*varname_e*) specifies a variable that reflects the amount of exposure over which the *depvar* events were observed for each observation; ln(*varname_e*) is included in the fixed-effects portion of the model with the coefficient constrained to be 1.

offset(*varname_o*) specifies that *varname_o* be included in the fixed-effects portion of the model with the coefficient constrained to be 1.

covariance(*vartype*) specifies the structure of the covariance matrix for the random effects and may be specified for each random-effects equation. *vartype* is one of the following: independent, exchangeable, identity, and unstructured.

covariance(independent) covariance structure allows for a distinct variance for each random effect within a random-effects equation and assumes that all covariances are 0. The default is covariance(independent), except when the R. notation is used, in which case the default is covariance(identity) and only covariance(identity) and covariance(exchangeable) are allowed.

covariance(exchangeable) structure specifies one common variance for all random effects and one common pairwise covariance.

covariance(identity) is short for "multiple of the identity"; that is, all variances are equal and all covariances are 0.

covariance(unstructured) allows for all variances and covariances to be distinct. If an equation consists of p random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ unique parameters.

collinear specifies that meqrpoisson not omit collinear variables from the random-effects equation. Usually, there is no reason to leave collinear variables in place; in fact, doing so usually causes the estimation to fail because of the matrix singularity caused by the collinearity. However, with certain models (for example, a random-effects model with a full set of contrasts), the variables may be collinear, yet the model is fully identified because of restrictions on the random-effects covariance structure. In such cases, using the collinear option allows the estimation to take place with the random-effects equation intact.

|‾ Reporting ‾|

level(*#*); see [R] **estimation options**.

irr reports estimated fixed-effects coefficients transformed to incidence-rate ratios, that is, $\exp(\beta)$ rather than β. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. irr may be specified at estimation or upon replay.

variance, the default, displays the random-effects parameter estimates as variances and covariances.

stddeviations displays the random-effects parameter estimates as standard deviations and correla-
tions.

noretable suppresses the random-effects table.

nofetable suppresses the fixed-effects table.

estmetric displays all parameter estimates in the estimation metric. Fixed-effects estimates are
unchanged from those normally displayed, but random-effects parameter estimates are displayed
as log-standard deviations and hyperbolic arctangents of correlations, with equation names that
organize them by model level.

noheader suppresses the output header, either at estimation or upon replay.

nogroup suppresses the display of group summary information (number of groups, average group
size, minimum, and maximum) from the output header.

nolrtest prevents meqrpoisson from performing a likelihood-ratio test that compares the mixed-
effects Poisson model with standard (marginal) Poisson regression. This option may also be specified
upon replay to suppress this test from the output.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels,
allbaselevels, nofvlabel, fvwrap(*#*), fvwrapon(*style*), cformat(*%fmt*), pformat(*%fmt*),
sformat(*%fmt*), and nolstretch; see [R] **estimation options**.

    Integration

intpoints(*#* [*#* ...]) sets the number of integration points for adaptive Gaussian quadrature. The
more integration points, the more accurate the approximation to the log likelihood. However,
computation time increases with the number of quadrature points, and in models with many levels
or many random coefficients, this increase can be substantial.

You may specify one number of integration points applying to all levels of random effects in
the model, or you may specify distinct numbers of points for each level. intpoints(7) is the
default; that is, by default seven quadrature points are used for each level.

laplace specifies that log likelihoods be calculated using the Laplacian approximation, equivalent
to adaptive Gaussian quadrature with one integration point for each level in the model; laplace
is equivalent to intpoints(1). Computation time increases as a function of the number of
quadrature points raised to a power equaling the dimension of the random-effects specification.
The computational time saved by using laplace can thus be substantial, especially when you
have many levels or random coefficients.

The Laplacian approximation has been known to produce biased parameter estimates, but the bias
tends to be more prominent in the estimates of the variance components rather than in the estimates
of the fixed effects. If your interest lies primarily with the fixed-effects estimates, the Laplace
approximation may be a viable faster alternative to adaptive quadrature with multiple integration
points.

When the R.*varname* notation is used, the dimension of the random effects increases by the
number of distinct values of *varname*. Even when this number is small to moderate, it increases
the total random-effects dimension to the point where estimation with more than one quadrature
point is prohibitively intensive.

For this reason, when you use the R. notation in your random-effects equations, the laplace
option is assumed. You can override this behavior by using the intpoints() option, but doing
so is not recommended.

⎡ Maximization ⎤

maximize_options: <u>d</u>ifficult, <u>tech</u>nique(*algorithm_spec*), <u>iter</u>ate(*#*), [<u>no</u>]<u>log</u>, <u>trace</u>,
  <u>grad</u>ient, <u>showstep</u>, <u>hess</u>ian, <u>showtol</u>erance, <u>tol</u>erance(*#*), <u>ltol</u>erance(*#*),
  <u>nrtol</u>erance(*#*), <u>nonrtol</u>erance, and <u>from</u>(*init_specs*); see [R] **maximize**. Those that require
  special mention for meqrpoisson are listed below.

  For the technique() option, the default is technique(nr). The bhhh algorithm may not be
  specified.

  from(*init_specs*) is particularly useful when combined with refineopts(iterate(0)) (see the
  description below), which bypasses the initial optimization stage.

<u>retol</u>erance(*#*) specifies the convergence tolerance for the estimated random effects used by adaptive
  Gaussian quadrature. Although not estimated as model parameters, random-effects estimators are
  used to adapt the quadrature points. Estimating these random effects is an iterative procedure,
  with convergence declared when the maximum relative change in the random effects is less than
  retolerance(). The default is retolerance(1e-8). You should seldom have to use this option.

<u>reiter</u>ate(*#*) specifies the maximum number of iterations used when estimating the random effects
  to be used in adapting the Gaussian quadrature points; see the retolerance() option. The default
  is reiterate(50). You should seldom have to use this option.

matsqrt (the default), during optimization, parameterizes variance components by using the matrix
  square roots of the variance–covariance matrices formed by these components at each model level.

matlog, during optimization, parameterizes variance components by using the matrix logarithms of
  the variance–covariance matrices formed by these components at each model level.

  The matsqrt parameterization ensures that variance–covariance matrices are positive semidefinite,
  while matlog ensures matrices that are positive definite. For most problems, the matrix square root
  is more stable near the boundary of the parameter space. However, if convergence is problematic,
  one option may be to try the alternate matlog parameterization. When convergence is not an issue,
  both parameterizations yield equivalent results.

<u>refineopts</u>(*maximize_options*) controls the maximization process during the refinement of starting
  values. Estimation in meqrpoisson takes place in two stages. In the first stage, starting values
  are refined by holding the quadrature points fixed between iterations. During the second stage,
  quadrature points are adapted with each evaluation of the log likelihood. Maximization options
  specified within refineopts() control the first stage of optimization; that is, they control the
  refining of starting values.

  maximize_options specified outside refineopts() control the second stage.

  The one exception to the above rule is the nolog option, which when specified outside refine-
  opts() applies globally.

  from(*init_specs*) is not allowed within refineopts() and instead must be specified globally.

  Refining starting values helps make the iterations of the second stage (those that lead toward the so-
  lution) more numerically stable. In this regard, of particular interest is refineopts(iterate(*#*)),
  with two iterations being the default. Should the maximization fail because of instability in the
  Hessian calculations, one possible solution may be to increase the number of iterations here.

The following option is available with meqrpoisson but is not shown in the dialog box:

<u>coefl</u>egend; see [R] **estimation options**.

Remarks and examples

Remarks are presented under the following headings:

Introduction

Mixed-effects Poisson regression is Poisson regression containing both fixed effects and random effects. In longitudinal data and panel data, random effects are useful for modeling intracluster correlation; that is, observations in the same cluster are correlated because they share common cluster-level random effects.

meqrpoisson allows for not just one, but many levels of nested clusters. For example, in a three-level model you can specify random effects for schools and then random effects for classes nested within schools. The observations (students, presumably) would comprise level one of the model, the classes would comprise level two, and the schools would comprise level three.

However, for simplicity, for now we consider the two-level model, where for a series of M independent clusters, and conditional on a set of random effects $\mathbf{u}_j$,

$$\Pr(y_{ij} = y | \mathbf{u}_j) = \exp\left(-\mu_{ij}\right) \mu_{ij}^y / y! \tag{1}$$

for $\mu_{ij} = \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$, $j = 1, \ldots, M$ clusters, and with cluster j consisting of $i = 1, \ldots, n_j$ observations. The responses are counts y_{ij}. The $1 \times p$ row vector $\mathbf{x}_{ij}$ are the covariates for the fixed effects, analogous to the covariates you would find in a standard Poisson regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$.

The $1 \times q$ vector $\mathbf{z}_{ij}$ are the covariates corresponding to the random effects and can be used to represent both random intercepts and random coefficients. For example, in a random-intercept model, $\mathbf{z}_{ij}$ is simply the scalar 1. The random effects $\mathbf{u}_j$ are M realizations from a multivariate normal distribution with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$. The random effects are not directly estimated as model parameters but are instead summarized according to the unique elements of $\boldsymbol{\Sigma}$, known as variance components. One special case of (1) places $\mathbf{z}_{ij} = \mathbf{x}_{ij}$ so that all covariate effects are essentially random and distributed as multivariate normal with mean $\boldsymbol{\beta}$ and variance $\boldsymbol{\Sigma}$.

Model (1) is an example of a generalized linear mixed model (GLMM), which generalizes the linear mixed-effects (LME) model to non-Gaussian responses. You can fit LMEs in Stata by using mixed and fit GLMMs by using meglm. Because of the relationship between LMEs and GLMMs, there is insight to be gained through examination of the linear mixed model. This is especially true for Stata users because the terminology, syntax, options, and output for fitting these types of models are nearly identical. See [ME] **mixed** and the references therein, particularly in the *Introduction*, for more information.

Log-likelihood calculations for fitting any generalized mixed-effects model require integrating out the random effects. One widely used modern method is to directly estimate the integral required to calculate the log likelihood by Gauss–Hermite quadrature or some variation thereof. The estimation method used by meqrpoisson is a multicoefficient and multilevel extension of one of these quadrature types, namely, adaptive Gaussian quadrature (AGQ) based on conditional modes, with the multicoefficient extension from Pinheiro and Bates (1995) and the multilevel extension from Pinheiro and Chao (2006); see *Methods and formulas*.

Below we present two short examples of mixed-effects Poisson regression; refer to [ME] **me** and [ME] **meglm** for additional examples.

A two-level model

In this section, we begin with a two-level mixed-effects Poisson regression, because a one-level model, in multilevel-model terminology, is just standard Poisson regression; see [R] **poisson**.

▷ Example 1

Breslow and Clayton (1993) fit a mixed-effects Poisson model to data from a randomized trial of the drug progabide for the treatment of epilepsy.

```
. use http://www.stata-press.com/data/r14/epilepsy
(Epilepsy data; progabide drug treatment)
. describe
Contains data from http://www.stata-press.com/data/r14/epilepsy.dta
  obs:            236                          Epilepsy data; progabide drug
                                                 treatment
  vars:             8                          31 May 2014 14:09
  size:         4,956                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
subject	byte	%9.0g		Subject ID: 1-59
seizures	int	%9.0g		No. of seizures
treat	byte	%9.0g		1: progabide; 0: placebo
visit	float	%9.0g		Dr. visit; coded as (-.3, -.1, .1, .3)
lage	float	%9.0g		log(age), mean-centered
lbas	float	%9.0g		log(0.25*baseline seizures), mean-centered
lbas_trt	float	%9.0g		lbas/treat interaction
v4	byte	%8.0g		Fourth visit indicator

```
Sorted by: subject
```

Originally from Thall and Vail (1990), data were collected on 59 subjects (31 on progabide, 28 on placebo). The number of epileptic seizures (seizures) was recorded during the two weeks prior to each of four doctor visits (visit). The treatment group is identified by the indicator variable treat. Data were also collected on the logarithm of age (lage) and the logarithm of one-quarter the number of seizures during the eight weeks prior to the study (lbas). The variable lbas_trt represents the interaction between lbas and treatment. lage, lbas, and lbas_trt are mean centered. Because the study originally noted a substantial decrease in seizures prior to the fourth doctor visit, an indicator, v4, for the fourth visit was also recorded.

Breslow and Clayton (1993) fit a random-effects Poisson model for the number of observed seizures

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \text{treat}_{ij} + \beta_2 \text{lbas}_{ij} + \beta_3 \text{lbas\_trt}_{ij} + \beta_4 \text{lage}_{ij} + \beta_5 \text{v4}_{ij} + u_j$$

for $j = 1, \ldots, 59$ subjects and $i = 1, \ldots, 4$ visits. The random effects u_j are assumed to be normally distributed with mean 0 and variance σ_u^2.

```
. meqrpoisson seizures treat lbas lbas_trt lage v4 || subject:

Refining starting values:

Iteration 0:    log likelihood = -680.40577  (not concave)
Iteration 1:    log likelihood = -668.60112
Iteration 2:    log likelihood = -666.37635

Performing gradient-based optimization:

Iteration 0:    log likelihood = -666.37635
Iteration 1:    log likelihood =  -665.4589
Iteration 2:    log likelihood = -665.29074
Iteration 3:    log likelihood = -665.29068
```

```
Mixed-effects Poisson regression              Number of obs      =        236
Group variable: subject                       Number of groups   =         59

                                              Obs per group:
                                                            min =          4
                                                            avg =        4.0
                                                            max =          4

Integration points =    7                     Wald chi2(5)       =     121.67
Log likelihood = -665.29068                   Prob > chi2        =     0.0000
```

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
treat	-.9330388	.4008346	-2.33	0.020	-1.71866	-.1474175
lbas	.884433	.1312313	6.74	0.000	.6272244	1.141642
lbas_trt	.3382609	.2033384	1.66	0.096	-.0602752	.7367969
lage	.4842389	.3472775	1.39	0.163	-.1964126	1.16489
v4	-.1610871	.0545758	-2.95	0.003	-.2680537	-.0541206
_cons	2.154574	.2200426	9.79	0.000	1.723299	2.58585

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
subject: Identity				
var(_cons)	.2528265	.058956	.1600785	.3993118

```
LR test vs. Poisson model: chibar2(01) = 304.74      Prob >= chibar2 = 0.0000
```

The number of seizures before the fourth visit does exhibit a significant drop, and the patients on progabide demonstrate a decrease in frequency of seizures compared with the placebo group. The subject-specific random effects also appear significant: $\hat{\sigma}_u^2 = 0.25$ with standard error 0.06. The above results are also in good agreement with those of Breslow and Clayton (1993, table 4), who fit this model by the method of penalized quasi-likelihood (PQL).

Because this is a simple random-intercept model, you can obtain equivalent results by using xtpoisson with the re and normal options.

◁

▷ Example 2

In their study of PQL, Breslow and Clayton (1993) also fit a model where they dropped the fixed effect on v4 and replaced it with a random subject-specific linear trend over the four doctor visits. The model they fit is

$$\log(\mu_{ij}) = \beta_0 + \beta_1 \texttt{treat}_{ij} + \beta_2 \texttt{lbas}_{ij} + \beta_3 \texttt{lbas\_trt}_{ij} +$$
$$\beta_4 \texttt{lage}_{ij} + \beta_5 \texttt{visit}_{ij} + u_j + v_j \texttt{visit}_{ij}$$

where (u_j, v_j) are bivariate normal with 0 mean and variance–covariance matrix

$$\boldsymbol{\Sigma} = \mathrm{Var}\begin{bmatrix} u_j \\ v_j \end{bmatrix} = \begin{bmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{bmatrix}$$

```
. meqrpoisson seizures treat lbas lbas_trt lage visit || subject: visit,
> cov(unstructured) intpoints(9)

Refining starting values:

Iteration 0:   log likelihood = -672.17188  (not concave)
Iteration 1:   log likelihood = -660.46056
Iteration 2:   log likelihood =  -655.8713

Performing gradient-based optimization:

Iteration 0:   log likelihood =  -655.8713
Iteration 1:   log likelihood = -655.68217
Iteration 2:   log likelihood = -655.68103
Iteration 3:   log likelihood = -655.68103
```

Mixed-effects Poisson regression				Number of obs	=	236
Group variable: subject				Number of groups	=	59

Obs per group:

				min =		4
				avg =		4.0
				max =		4

Integration points = 9				Wald chi2(5)	=	115.56
Log likelihood = -655.68103				Prob > chi2	=	0.0000

seizures	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
treat	-.9286589	.4021642	-2.31	0.021	-1.716886	-.1404315
lbas	.8849766	.131252	6.74	0.000	.6277275	1.142226
lbas_trt	.3379758	.2044444	1.65	0.098	-.0627279	.7386794
lage	.4767192	.3536221	1.35	0.178	-.2163673	1.169806
visit	-.2664098	.1647096	-1.62	0.106	-.5892347	.0564151
_cons	2.099555	.2203711	9.53	0.000	1.667635	2.531474

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
subject: Unstructured				
var(visit)	.5314808	.2293851	.2280931	1.238406
var(_cons)	.2514928	.0587892	.1590552	.3976522
cov(visit,_cons)	.0028715	.0887018	-.1709808	.1767238

LR test vs. Poisson model: chi2(3) = 324.54   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.

In the above, we specified the cov(unstructured) option to allow correlation between u_j and v_j, although on the basis of the above output it probably was not necessary—the default Independent structure would have sufficed. In the interest of getting more accurate estimates, we also increased the number of quadrature points to nine, although the estimates do not change much when compared with estimates based on the default seven quadrature points.

The essence of the above-fitted model is that after adjusting for other covariates, the log trend in seizures is modeled as a random subject-specific line, with intercept distributed as $N(\beta_0, \sigma_u^2)$ and slope distributed as $N(\beta_5, \sigma_v^2)$. From the above output, $\widehat{\beta}_0 = 2.10$, $\widehat{\sigma}_u^2 = 0.25$, $\widehat{\beta}_5 = -0.27$, and $\widehat{\sigma}_v^2 = 0.53$.

You can predict the random effects u_j and v_j by using predict after meqrpoisson; see [ME] **meqrpoisson postestimation**. Better still, you can obtain a predicted number of seizures that takes these random effects into account.

◁

A three-level model

meqrpoisson can also fit higher-level models with multiple levels of nested random effects.

▷ Example 3

Rabe-Hesketh and Skrondal (2012, exercise 13.7) describe data from the *Atlas of Cancer Mortality in the European Economic Community* (EEC) (Smans, Mair, and Boyle 1993). The data were analyzed in Langford, Bentham, and McDonald (1998) and record the number of deaths among males due to malignant melanoma during 1971–1980.

```
. use http://www.stata-press.com/data/r14/melanoma
(Skin cancer (melanoma) data)

. describe
Contains data from http://www.stata-press.com/data/r14/melanoma.dta
  obs:           354                          Skin cancer (melanoma) data
 vars:             6                          30 May 2014 17:10
 size:         4,956                          (_dta has notes)
```

variable name	storage type	display format	value label	variable label
nation	byte	%11.0g	n	Nation ID
region	byte	%9.0g		Region ID: EEC level-I areas
county	int	%9.0g		County ID: EEC level-II/level-III areas
deaths	int	%9.0g		No. deaths during 1971–1980
expected	float	%9.0g		No. expected deaths
uv	float	%9.0g		UV dose, mean-centered

```
Sorted by:
```

Nine European nations (variable nation) are represented, and data were collected over geographical regions defined by EEC statistical services as level I areas (variable region), with deaths being recorded for each of 354 counties, which are level II or level III EEC-defined areas (variable county, which identifies the observations). Counties are nested within regions, and regions are nested within nations.

The variable deaths records the number of deaths for each county, and expected records the expected number of deaths (the exposure) on the basis of crude rates for the combined countries. Finally, the variable uv is a measure of exposure to ultraviolet (UV) radiation.

In modeling the number of deaths, one possibility is to include dummy variables for the nine nations as fixed effects. Another is to treat these as random effects and fit the three-level random-intercept Poisson model,

$$\log(\mu_{ijk}) = \log(\texttt{expected}_{ijk}) + \beta_0 + \beta_1 \texttt{uv}_{ijk} + u_k + v_{jk}$$

for nation k, region j, and county i. The model includes an exposure term for expected deaths.

```
. meqrpoisson deaths uv, exposure(expected) || nation: || region:

Refining starting values:

Iteration 0:    log likelihood = -1169.0851   (not concave)
Iteration 1:    log likelihood =  -1156.523   (not concave)
Iteration 2:    log likelihood = -1101.8488

Performing gradient-based optimization:

Iteration 0:    log likelihood = -1101.8488
Iteration 1:    log likelihood = -1099.5579
Iteration 2:    log likelihood = -1097.8162
Iteration 3:    log likelihood = -1097.7151
Iteration 4:    log likelihood =  -1097.714
Iteration 5:    log likelihood =  -1097.714
```

Mixed-effects Poisson regression   Number of obs   =   354

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
nation	9	3	39.3	95	7
region	78	1	4.5	13	7

Log likelihood = -1097.714

Wald chi2(1)   =   6.12
Prob > chi2   =   0.0134

deaths	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
uv	-.0281991	.0114027	-2.47	0.013	-.050548	-.0058503
_cons	-.0639472	.1335247	-0.48	0.632	-.3256508	.1977565
ln(expected)	1	(exposure)				

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]	
nation: Identity				
var(_cons)	.1370344	.0722804	.0487364	.3853059
region: Identity				
var(_cons)	.0483854	.010927	.0310803	.0753258

LR test vs. Poisson model: chi2(2) = 1252.12   Prob > chi2 = 0.0000

Note: LR test is conservative and provided only for reference.

By including an exposure variable that is an expected rate, we are in effect specifying a linear model for the log of the standardized mortality ratio, the ratio of observed deaths to expected deaths that is based on a reference population. Here the reference population is all nine nations.

We now add a random intercept for counties nested within regions, making this a four-level model. Because counties also identify the observations, the corresponding variance component can be interpreted as a measure of overdispersion, variability above and beyond that allowed by a Poisson process; see [R] **nbreg** and [ME] **menbreg**.

```
. meqrpoisson deaths uv, exposure(expected) || nation: || region: || county:,
> laplace

Refining starting values:

Iteration 0:   log likelihood = -1381.1202  (not concave)
Iteration 1:   log likelihood = -1144.7025  (not concave)
Iteration 2:   log likelihood = -1138.7095

Performing gradient-based optimization:

Iteration 0:   log likelihood = -1138.7095
Iteration 1:   log likelihood = -1129.8553
Iteration 2:   log likelihood = -1097.7408  (not concave)
Iteration 3:   log likelihood = -1088.4381
Iteration 4:   log likelihood = -1086.8459
Iteration 5:   log likelihood = -1086.7321
Iteration 6:   log likelihood = -1086.7309
Iteration 7:   log likelihood = -1086.7309
```

```
Mixed-effects Poisson regression                  Number of obs     =        354
```

Group Variable	No. of Groups	Observations per Group Minimum	Average	Maximum	Integration Points
nation	9	3	39.3	95	1
region	78	1	4.5	13	1
county	354	1	1.0	1	1

```
                                                  Wald chi2(1)      =       8.63
Log likelihood = -1086.7309                       Prob > chi2       =     0.0033
```

deaths	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]
uv	-.0334681	.0113919	-2.94	0.003	-.0557959 -.0111404
_cons	-.0864109	.1298712	-0.67	0.506	-.3409538 .168132
ln(expected)	1	(exposure)			

Random-effects Parameters	Estimate	Std. Err.	[95% Conf. Interval]
nation: Identity			
var(_cons)	.1287416	.0680887	.0456601 .3629954
region: Identity			
var(_cons)	.0405965	.0105002	.0244527 .0673986
county: Identity			
var(_cons)	.0146027	.0050766	.0073878 .0288636

```
LR test vs. Poisson model: chi2(3) = 1274.08         Prob > chi2 = 0.0000
```

Note: LR test is conservative and provided only for reference.
Note: Log-likelihood calculations are based on the Laplacian approximation.

In the above, we used a Laplacian approximation, which is not only faster but also produces estimates that closely agree with those obtained with the default seven quadrature points.

See *Computation time and the Laplacian approximation* in [ME] **me** for a discussion comparing Laplacian approximation with adaptive quadrature.

◁

Stored results

meqrpoisson stores the following in e():

Scalars
e(N)	number of observations
e(k)	number of parameters
e(k_f)	number of fixed-effects parameters
e(k_r)	number of random-effects parameters
e(k_rs)	number of variances
e(k_rc)	number of covariances
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(chi2)	χ^2
e(p)	significance
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison model
e(df_c)	degrees of freedom, comparison model
e(p_c)	significance, comparison model
e(rank)	rank of e(V)
e(reparm_rc)	return code, final reparameterization
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros
e(cmd)	meqrpoisson
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(ivars)	grouping variables
e(exposurevar)	exposure variable
e(model)	Poisson
e(title)	title in estimation output
e(offset)	offset
e(redim)	random-effects dimensions
e(vartypes)	variance-structure types
e(revars)	random-effects covariates
e(n_quad)	number of integration points
e(laplace)	laplace, if Laplace approximation
e(chi2type)	Wald; type of model χ^2
e(vce)	bootstrap or jackknife if defined
e(vcetype)	title used to label Std. Err.
e(method)	ML
e(opt)	type of optimization
e(ml_method)	type of ml method
e(technique)	maximization technique
e(datasignature)	the checksum
e(datasignaturevars)	variables used in calculation of checksum
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(marginsnotok)	predictions disallowed by margins
e(marginsdefault)	default predict() specification for margins
e(asbalanced)	factor variables fvset as asbalanced
e(asobserved)	factor variables fvset as asobserved

Matrices
e(b)	coefficient vector
e(Cns)	constraints matrix
e(N_g)	group counts
e(g_min)	group-size minimums
e(g_avg)	group-size averages
e(g_max)	group-size maximums
e(V)	variance–covariance matrix of the estimators

Functions
e(sample)	marks estimation sample

Methods and formulas

In a two-level Poisson model, for cluster j, $j = 1, \ldots, M$, the conditional distribution of $\mathbf{y}_j = (y_{j1}, \ldots, y_{jn_j})'$, given a set of cluster-level random effects $\mathbf{u}_j$, is

$$
f(\mathbf{y}_j|\mathbf{u}_j) = \prod_{i=1}^{n_j} \left[\{ \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) \}^{y_{ij}} \exp\{ -\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) \} / y_{ij}! \right]
$$

$$
= \exp\left[\sum_{i=1}^{n_j} \{ y_{ij}(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) - \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j) - \log(y_{ij}!) \} \right]
$$

Defining $c(\mathbf{y}_j) = \sum_{i=1}^{n_j} \log(y_{ij}!)$, where $c(\mathbf{y}_j)$ does not depend on the model parameters, we can express the above compactly in matrix notation,

$$
f(\mathbf{y}_j|\mathbf{u}_j) = \exp\{ \mathbf{y}_j'(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \mathbf{1}'\exp(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - c(\mathbf{y}_j) \}
$$

where $\mathbf{X}_j$ is formed by stacking the row vectors $\mathbf{x}_{ij}$ and $\mathbf{Z}_j$ is formed by stacking the row vectors $\mathbf{z}_{ij}$. We extend the definition of $\exp(\cdot)$ to be a vector function where necessary.

Because the prior distribution of $\mathbf{u}_j$ is multivariate normal with mean $\mathbf{0}$ and $q \times q$ variance matrix $\boldsymbol{\Sigma}$, the likelihood contribution for the jth cluster is obtained by integrating $\mathbf{u}_j$ out of the joint density $f(\mathbf{y}_j, \mathbf{u}_j)$,

$$
\mathcal{L}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = (2\pi)^{-q/2} |\boldsymbol{\Sigma}|^{-1/2} \int f(\mathbf{y}_j|\mathbf{u}_j) \exp\left(-\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2 \right) d\mathbf{u}_j
$$

$$
= \exp\{ -c(\mathbf{y}_j) \} (2\pi)^{-q/2} |\boldsymbol{\Sigma}|^{-1/2} \int \exp\{ h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) \} d\mathbf{u}_j
$$

(2)

where

$$
h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \mathbf{y}_j'(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \mathbf{1}'\exp(\mathbf{X}_j\boldsymbol{\beta} + \mathbf{Z}_j\mathbf{u}_j) - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2
$$

and for convenience, in the arguments of $h(\cdot)$ we suppress the dependence on the observable data $(\mathbf{y}_j, \mathbf{X}_j, \mathbf{Z}_j)$.

The integration in (2) has no closed form and thus must be approximated. The Laplacian approximation (Tierney and Kadane 1986; Pinheiro and Bates 1995) is based on a second-order Taylor expansion of $h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ about the value of $\mathbf{u}_j$ that maximizes it. Taking first and second derivatives, we obtain

$$
h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \frac{\partial h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)}{\partial \mathbf{u}_j} = \mathbf{Z}_j'\{ \mathbf{y}_j - \mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j) \} - \boldsymbol{\Sigma}^{-1}\mathbf{u}_j
$$

$$
h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) = \frac{\partial^2 h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)}{\partial \mathbf{u}_j \partial \mathbf{u}_j'} = -\{ \mathbf{Z}_j'\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)\mathbf{Z}_j + \boldsymbol{\Sigma}^{-1} \}
$$

where $\mathbf{m}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the vector function with the ith element equal to the conditional mean of y_{ij} given $\mathbf{u}_j$, that is, $\exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$. $\mathbf{V}(\boldsymbol{\beta}, \mathbf{u}_j)$ is the diagonal matrix whose diagonal entries v_{ij} are the conditional variances of y_{ij} given $\mathbf{u}_j$, namely,

$$
v_{ij} = \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)
$$

because equality of mean and variance is a characteristic of the Poisson distribution.

The maximizer of $h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)$ is $\widehat{\mathbf{u}}_j$ such that $h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. The integrand in (2) is proportional to the posterior density $f(\mathbf{u}_j|\mathbf{y}_j)$, so $\widehat{\mathbf{u}}_j$ also represents the posterior mode, a plausible estimator of $\mathbf{u}_j$ in its own right.

Given the above derivatives, the second-order Taylor approximation then takes the form

$$h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j) \approx h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) + \frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)' h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)(\mathbf{u}_j - \widehat{\mathbf{u}}_j) \tag{3}$$

The first-derivative term vanishes because $h'(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{0}$. Therefore,

$$\begin{aligned}
\int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\} \, d\mathbf{u}_j &\approx \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\} \\
&\times \int \exp\left[-\frac{1}{2}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)'\{-h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\}(\mathbf{u}_j - \widehat{\mathbf{u}}_j)\right] d\mathbf{u}_j \\
&= \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\}(2\pi)^{q/2} \left|-h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j)\right|^{-1/2}
\end{aligned} \tag{4}$$

because the latter integrand can be recognized as the "kernel" of a multivariate normal density.

Combining the above with (2) (and taking logs) gives the Laplacian log-likelihood contribution of the jth cluster,

$$\mathcal{L}_j^{\mathrm{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| - \log|\mathbf{R}_j| + h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) - c(\mathbf{y}_j)$$

where $\mathbf{R}_j$ is an upper-triangular matrix such that $-h''(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j) = \mathbf{R}_j \mathbf{R}_j'$. Pinheiro and Chao (2006) show that $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j$ can be efficiently computed as the iterative solution to a least-squares problem by using matrix decomposition methods similar to those used in fitting LME models (Bates and Pinheiro 1998; Pinheiro and Bates 2000; [ME] **mixed**).

The fidelity of the Laplacian approximation is determined wholly by the accuracy of the approximation in (3). An alternative that does not depend so heavily on this approximation is integration via AGQ (Naylor and Smith 1982; Liu and Pierce 1994).

The application of AGQ to this particular problem is from Pinheiro and Bates (1995). When we reexamine the integral in question, a transformation of integration variables yields

$$\begin{aligned}
\int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j)\} \, d\mathbf{u}_j &= |\mathbf{R}_j|^{-1} \int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t})\} \, d\mathbf{t} \\
&= (2\pi)^{q/2} |\mathbf{R}_j|^{-1} \int \exp\{h(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{t}) + \mathbf{t}'\mathbf{t}/2\} \phi(\mathbf{t}) d\mathbf{t}
\end{aligned} \tag{5}$$

where $\phi(\cdot)$ is the standard multivariate normal density. Because the integrand is now expressed as some function multiplied by a normal density, it can be estimated by applying the rules of standard Gauss–Hermite quadrature. For a predetermined number of quadrature points N_Q, define $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$, for $k = 1, \ldots, N_Q$, where (a_k^*, w_k^*) are a set of abscissas and weights for Gauss–Hermite quadrature approximations of $\int \exp(-x^2)f(x)dx$, as obtained from Abramowitz and Stegun (1972, 924).

Define $\mathbf{a_k} = (a_{k_1}, a_{k_2}, \ldots, a_{k_q})'$; that is, $\mathbf{a_k}$ is a vector that spans the N_Q abscissas over the dimension q of the random effects. Applying quadrature rules to (5) yields the AGQ approximation,

$$\int \exp\left\{h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \mathbf{u}_j\right)\right\} d\mathbf{u}_j$$

$$\approx (2\pi)^{q/2} \left|\mathbf{R}_j\right|^{-1} \sum_{k_1=1}^{N_Q} \cdots \sum_{k_q=1}^{N_Q} \left[\exp\left\{h\left(\boldsymbol{\beta}, \boldsymbol{\Sigma}, \widehat{\mathbf{u}}_j + \mathbf{R}_j^{-1}\mathbf{a_k}\right) + \mathbf{a_k}'\mathbf{a_k}/2\right\} \prod_{p=1}^{q} w_{k_p} \right]$$

$$\equiv (2\pi)^{q/2} \widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})$$

resulting in the AGQ log-likelihood contribution of the jth cluster,

$$\mathcal{L}_j^{\text{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}| + \log\left\{\widehat{G}_j(\boldsymbol{\beta}, \boldsymbol{\Sigma})\right\} - c(\mathbf{y}_j)$$

The "adaptive" part of adaptive Gaussian quadrature lies in the translation and rescaling of the integration variables in (5) by using $\widehat{\mathbf{u}}_j$ and $\mathbf{R}_j^{-1}$, respectively. This transformation of quadrature abscissas (centered at 0 in standard form) is chosen to better capture the features of the integrand, through which (4) can be seen to resemble a multivariate normal distribution with mean $\widehat{\mathbf{u}}_j$ and variance $\mathbf{R}_j^{-1}\mathbf{R}_j^{-T}$. AGQ is therefore not as dependent as the Laplace method upon the approximation in (3). In AGQ, (3) serves merely to redirect the quadrature abscissas, with the AGQ approximation improving as the number of quadrature points, N_Q, increases. In fact, Pinheiro and Bates (1995) point out that AGQ with only one quadrature point ($a = 0$ and $w = 1$) reduces to the Laplacian approximation.

The log likelihood for the entire dataset is then simply the sum of the contributions of the M individual clusters, namely, $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} \mathcal{L}_j^{\text{Lap}}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ for Laplace and $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{j=1}^{M} \mathcal{L}_j^{\text{AGQ}}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ for AGQ.

Maximization of $\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$ is performed with respect to $(\boldsymbol{\beta}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector comprising the unique elements of the matrix square root of $\boldsymbol{\Sigma}$. This is done to ensure that $\boldsymbol{\Sigma}$ is always positive semidefinite. If the `matlog` option is specified, then $\boldsymbol{\theta}$ instead consists of the unique elements of the matrix logarithm of $\boldsymbol{\Sigma}$. For well-conditioned problems, both methods produce equivalent results, yet our experience deems the former as more numerically stable near the boundary of the parameter space.

Once maximization is achieved, parameter estimates are mapped from $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\theta}})$ to $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, where $\widehat{\boldsymbol{\gamma}}$ is a vector containing the unique (estimated) elements of $\boldsymbol{\Sigma}$, expressed as logarithms of standard deviations for the diagonal elements and hyperbolic arctangents of the correlations for off-diagonal elements. This last step is necessary to (a) obtain a parameterization under which parameter estimates can be displayed and interpreted individually, rather than as elements of a matrix square root (or logarithm), and (b) parameterize these elements such that their ranges each encompass the entire real line.

Parameter estimates are stored in `e(b)` as $(\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}})$, with the corresponding variance–covariance matrix stored in `e(V)`. Parameter estimates can be displayed in this metric by specifying the `estmetric` option. However, in `meqrpoisson` output, variance components are most often displayed either as variances and covariances (the default) or as standard deviations and correlations (option `stddeviations`).

The approach outlined above can be extended from two-level models to models with three or more levels; see Pinheiro and Chao (2006) for details.

References

Abramowitz, M., and I. A. Stegun, ed. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* 10th ed. Washington, DC: National Bureau of Standards.

Andrews, M. J., T. Schank, and R. Upward. 2006. Practical fixed-effects estimation methods for the three-way error-components model. *Stata Journal* 6: 461–481.

Bates, D. M., and J. C. Pinheiro. 1998. Computational methods for multilevel modelling. In *Technical Memorandum BL0112140-980226-01TM.* Murray Hill, NJ: Bell Labs, Lucent Technologies. http://stat.bell-labs.com/NLME/CompMulti.pdf.

Breslow, N. E., and D. G. Clayton. 1993. Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88: 9–25.

Gutierrez, R. G., S. L. Carter, and D. M. Drukker. 2001. sg160: On boundary-value likelihood-ratio tests. *Stata Technical Bulletin* 60: 15–18. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 269–273. College Station, TX: Stata Press.

Joe, H. 2008. Accuracy of Laplace approximation for discrete response mixed models. *Computational Statistics & Data Analysis* 52: 5066–5074.

Laird, N. M., and J. H. Ware. 1982. Random-effects models for longitudinal data. *Biometrics* 38: 963–974.

Langford, I. H., G. Bentham, and A. McDonald. 1998. Multi-level modelling of geographically aggregated health data: A case study on malignant melanoma mortality and UV exposure in the European community. *Statistics in Medicine* 17: 41–57.

Leyland, A. H., and H. Goldstein, ed. 2001. *Multilevel Modelling of Health Statistics.* New York: Wiley.

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

Liu, Q., and D. A. Pierce. 1994. A note on Gauss–Hermite quadrature. *Biometrika* 81: 624–629.

Marchenko, Y. V. 2006. Estimating variance components in Stata. *Stata Journal* 6: 1–21.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus. 2008. *Generalized, Linear, and Mixed Models.* 2nd ed. Hoboken, NJ: Wiley.

McLachlan, G. J., and K. E. Basford. 1988. *Mixture Models: Inference and Applications to Clustering.* New York: Dekker.

Naylor, J. C., and A. F. M. Smith. 1982. Applications of a method for the efficient computation of posterior distributions. *Journal of the Royal Statistical Society, Series C* 31: 214–225.

Palmer, T. M., C. M. Macdonald-Wallis, D. A. Lawlor, and K. Tilling. 2014. Estimating adjusted associations between random effects from multilevel models: The reffadjust package. *Stata Journal* 14: 119–140.

Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the log-likelihood function in the nonlinear mixed-effects model. *Journal of Computational and Graphical Statistics* 4: 12–35.

——. 2000. *Mixed-Effects Models in S and S-PLUS.* New York: Springer.

Pinheiro, J. C., and E. C. Chao. 2006. Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models. *Journal of Computational and Graphical Statistics* 15: 58–81.

Rabe-Hesketh, S., and A. Skrondal. 2012. *Multilevel and Longitudinal Modeling Using Stata.* 3rd ed. College Station, TX: Stata Press.

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

Raudenbush, S. W., and A. S. Bryk. 2002. *Hierarchical Linear Models: Applications and Data Analysis Methods.* 2nd ed. Thousand Oaks, CA: Sage.

Self, S. G., and K.-Y. Liang. 1987. Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions. *Journal of the American Statistical Association* 82: 605–610.

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

Smans, M., C. S. Mair, and P. Boyle. 1993. *Atlas of Cancer Mortality in the European Economic Community.* Lyon, France: IARC Scientific Publications.

Thall, P. F., and S. C. Vail. 1990. Some covariance models for longitudinal count data with overdispersion. *Biometrics* 46: 657–671.

Tierney, L., and J. B. Kadane. 1986. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81: 82–86.

Also see

[ME] **meqrpoisson postestimation** — Postestimation tools for meqrpoisson

[ME] **menbreg** — Multilevel mixed-effects negative binomial regression

[ME] **mepoisson** — Multilevel mixed-effects Poisson regression

[ME] **me** — Introduction to multilevel mixed-effects models

[MI] **estimation** — Estimation commands for use with mi estimate

[SEM] **intro 5** — Tour of models (*Multilevel mixed-effects models*)

[XT] **xtpoisson** — Fixed-effects, random-effects, and population-averaged Poisson models

[U] **20 Estimation and postestimation commands**