Title

stata.com

menbreg — Multilevel mixed-effects negative binomial regression

Description

`menbreg` fits mixed-effects negative binomial models to count data. The conditional distribution of the response given random effects is assumed to follow a Poisson-like process, except that the variation is greater than that of a true Poisson process.

Quick start

Mixed-effects negative binomial regression of `y` on `x` with random intercepts by `v1`

```
menbreg y x || v1:
```

Add `evar` measuring exposure

```
menbreg y x, exposure(evar) || v1:
```

As above, but report incidence-rate ratios instead of coefficients

```
menbreg y x, exposure(evar) || v1:, irr
```

Add random coefficients for `x`

```
menbreg y x, exposure(evar) || v1: x, irr
```

Three-level random-intercept model of `y` on `x` with `v1` nested within `v2`

```
menbreg y x || v2: || v1:
```

Menu

Statistics > Multilevel mixed-effects models > Negative binomial regression

Syntax

```
menbreg depvar fe_equation [ || re_equation ] [ || re_equation ... ] [ , options ]
```

where the syntax of *fe_equation* is

```
[ indepvars ] [ if ] [ in ] [ weight ] [ , 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
```

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	
`noconstant`	suppress the constant term from the fixed-effects equation
`exposure(`*varname*$_e$`)`	include $\ln($*varname*$_e)$ in model with coefficient constrained to 1
`offset(`*varname*$_o$`)`	include *varname*$_o$ in model with coefficient constrained to 1

re_options	Description
Model	
`covariance(`*vartype*`)`	variance–covariance structure of the random effects
`noconstant`	suppress constant term from the random-effects equation
`fweight(`*varname*`)`	frequency weights at higher levels
`iweight(`*varname*`)`	importance weights at higher levels
`pweight(`*varname*`)`	sampling weights at higher levels

options	Description
Model	
`dispersion(`*dispersion*`)`	parameterization of the conditional overdispersion; *dispersion* may be `mean` (default) or `constant`
`constraints(`*constraints*`)`	apply specified linear constraints
`collinear`	keep collinear variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, or `cluster` *clustvar*
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`irr`	report fixed-effects coefficients as incidence-rate ratios
`nocnsreport`	do not display constraints
`notable`	suppress coefficient table
`noheader`	suppress output header
`nogroup`	suppress table summarizing groups
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	
`intmethod(`*intmethod*`)`	integration method
`intpoints(#)`	set the number of integration (quadrature) points for all levels; default is `intpoints(7)`
Maximization	
maximize_options	control the maximization process; seldom used
`startvalues(`*svmethod*`)`	method for obtaining starting values
`startgrid`[`(`*gridspec*`)`]	perform a grid search to improve starting values
`noestimate`	do not fit the model; show starting values instead
`dnumerical`	use numerical derivative techniques
`coeflegend`	display legend instead of statistics

vartype	Description
`independent`	one unique variance parameter per random effect, all covariances 0; the default unless the `R.` notation is used
`exchangeable`	equal variances for random effects, and one common pairwise covariance
`identity`	equal variances for random effects, all covariances 0; the default if the `R.` notation is used
`unstructured`	all variances and covariances to be distinctly estimated
`fixed(`*matname*`)`	user-selected variances and covariances constrained to specified values; the remaining variances and covariances unrestricted
`pattern(`*matname*`)`	user-selected variances and covariances constrained to be equal; the remaining variances and covariances unrestricted

intmethod	Description
mvaghermite	mean–variance adaptive Gauss–Hermite quadrature; the default unless a crossed random-effects model is fit
mcaghermite	mode-curvature adaptive Gauss–Hermite quadrature
ghermite	nonadaptive Gauss–Hermite quadrature
laplace	Laplacian approximation; the default for crossed random-effects models

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

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

`bayes`, `by`, and `svy` are allowed; see [U] **11.1.10 Prefix commands**. For more details, see [BAYES] **bayes: menbreg**.

`vce()` and weights are not allowed with the `svy` prefix; see [SVY] **svy**.

`fweights`, `iweights`, and `pweights` are allowed; see [U] **11.1.6 weight**. Only one type of weight may be specified. Weights are not supported under the Laplacian approximation or for crossed models.

`startvalues()`, `startgrid`, `noestimate`, `dnumerical`, and `coeflegend` do 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`, `unstructured`, `fixed(`*matname*`)`, or `pattern(`*matname*`)`.

`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)` unless a crossed random-effects model is fit, in which case the default is `covariance(identity)`.

`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.

`covariance(fixed(`*matname*`))` and `covariance(pattern(`*matname*`))` covariance structures provide a convenient way to impose constraints on variances and covariances of random effects. Each specification requires a *matname* that defines the restrictions placed on variances and covariances. Only elements in the lower triangle of *matname* are used, and row and column names

of *matname* are ignored. A missing value in *matname* means that a given element is unrestricted. In a `fixed(`*matname*`)` covariance structure, (co)variance (i, j) is constrained to equal the value specified in the i, jth entry of *matname*. In a `pattern(`*matname*`)` covariance structure, (co)variances (i, j) and (k, l) are constrained to be equal if *matname*$[i, j] =$ *matname*$[k, l]$.

`fweight(`*varname*`)` specifies frequency weights at higher levels in a multilevel model, whereas frequency weights at the first level (the observation level) are specified in the usual manner, for example, `[fw=`*fwtvar1*`]`. *varname* can be any valid Stata variable name, and you can specify `fweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mecmd fixed_portion [fw = wt1] || school: ... , fweight(wt2) ...
```

the variable `wt1` would hold the first-level (the observation-level) frequency weights, and `wt2` would hold the second-level (the school-level) frequency weights.

`iweight(`*varname*`)` specifies importance weights at higher levels in a multilevel model, whereas importance weights at the first level (the observation level) are specified in the usual manner, for example, `[iw=`*iwtvar1*`]`. *varname* can be any valid Stata variable name, and you can specify `iweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mecmd fixed_portion [iw = wt1] || school: ... , iweight(wt2) ...
```

the variable `wt1` would hold the first-level (the observation-level) importance weights, and `wt2` would hold the second-level (the school-level) importance weights.

`pweight(`*varname*`)` specifies sampling weights at higher levels in a multilevel model, whereas sampling weights at the first level (the observation level) are specified in the usual manner, for example, `[pw=`*pwtvar1*`]`. *varname* can be any valid Stata variable name, and you can specify `pweight()` at levels two and higher of a multilevel model. For example, in the two-level model

```
. mecmd fixed_portion [pw = wt1] || school: ... , pweight(wt2) ...
```

variable `wt1` would hold the first-level (the observation-level) sampling weights, and `wt2` would hold the second-level (the school-level) sampling weights.

`dispersion(mean`|`constant)` specifies the parameterization of the conditional overdispersion given random effects. `dispersion(mean)`, the default, yields a model where the conditional overdispersion is a function of the conditional mean given random effects. For example, in a two-level model, the conditional overdispersion is equal to $1+\alpha E(y_{ij}|\mathbf{u}_j)$. `dispersion(constant)` yields a model where the conditional overdispersion is constant and is equal to $1 + \delta$. α and δ are the respective conditional overdispersion parameters.

`constraints(`*constraints*`)`, `collinear`; see [R] **estimation options**.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`), that are robust to some kinds of misspecification (`robust`), and that allow for intragroup correlation (`cluster` *clustvar*); see [R] ***vce_option***. If `vce(robust)` is specified, robust variances are clustered at the highest level in the multilevel model.

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 or stored. irr may be specified either at estimation or upon replay.

nocnsreport; see [R] estimation options.

notable suppresses the estimation table, either at estimation or upon replay.

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.

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

intmethod(*intmethod*) specifies the integration method to be used for the random-effects model. mvaghermite performs mean–variance adaptive Gauss–Hermite quadrature; mcaghermite performs mode-curvature adaptive Gauss–Hermite quadrature; ghermite performs nonadaptive Gauss–Hermite quadrature; and laplace performs the Laplacian approximation, equivalent to mode-curvature adaptive Gaussian quadrature with one integration point.

The default integration method is mvaghermite unless a crossed random-effects model is fit, in which case the default integration method is laplace. The Laplacian approximation has been known to produce biased parameter estimates; however, the bias tends to be more prominent in the estimates of the variance components rather than in the estimates of the fixed effects.

For crossed random-effects models, estimation with more than one quadrature point may be prohibitively intensive even for a small number of levels. For this reason, the integration method defaults to the Laplacian approximation. You may override this behavior by specifying a different integration method.

intpoints(#) sets the number of integration points for quadrature. The default is intpoints(7), which means that seven quadrature points are used for each level of random effects. This option is not allowed with intmethod(laplace).

The more integration points, the more accurate the approximation to the log likelihood. However, computation time increases as a function of the number of quadrature points raised to a power equaling the dimension of the random-effects specification. In crossed random-effects models and in models with many levels or many random coefficients, this increase can be substantial.

Maximization

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(*init_specs*); see [R] maximize. Those that require special mention for menbreg are listed below.

from() accepts a properly labeled vector of initial values or a list of coefficient names with values. A list of values is not allowed.

The following options are available with menbreg but are not shown in the dialog box:

startvalues(*svmethod*), startgrid[(*gridspec*)], noestimate, and dnumerical; see [ME] meglm.

coeflegend; see [R] estimation options.

Remarks and examples

stata.com

Mixed-effects negative binomial regression is negative binomial 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.

menbreg allows for many levels of random effects. However, for simplicity, consider a two-level model, where for a series of M independent clusters, and conditional on the latent variable ζ_{ij} and a set of random effects $\mathbf{u}_j$,

$$y_{ij}|\zeta_{ij} \sim \text{Poisson}(\zeta_{ij})$$

and

$$\zeta_{ij}|\mathbf{u}_j \sim \text{Gamma}(r_{ij}, p_{ij})$$

and

$$\mathbf{u}_j \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

where y_{ij} is the count response of the ith observation, $i = 1, \ldots, n_j$, from the jth cluster, $j = 1, \ldots, M$, and r_{ij} and p_{ij} have two different parameterizations, (2) and (3) below. 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.

The probability that a random response y_{ij} takes the value y is then given by

$$\Pr(y_{ij} = y|\mathbf{u}_j) = \frac{\Gamma(y + r_{ij})}{\Gamma(y+1)\Gamma(r_{ij})} p_{ij}^{r_{ij}} (1 - p_{ij})^y \tag{1}$$

where for convenience we suppress the dependence of the observable data y_{ij} on r_{ij} and p_{ij}.

Model (1) is an extension of the standard negative binomial model (see [R] nbreg) to incorporate normally distributed random effects at different hierarchical levels. (The negative binomial model itself can be viewed as a random-effects model, a Poisson model with a gamma-distributed random effect.) The standard negative binomial model is used to model overdispersed count data for which the variance is greater than that of a Poisson model. In a Poisson model, the variance is equal to the mean, and thus overdispersion is defined as the extra variability compared with the mean. According to this definition, the negative binomial model presents two different parameterizations of the overdispersion: the mean parameterization, where the overdispersion is a function of the mean, $1 + \alpha E(Y|\mathbf{x}), \alpha > 0$; and the constant parameterization, where the overdispersion is a constant function, $1 + \delta, \delta \geq 0$. We refer to α and δ as conditional overdispersion parameters.

Let $\mu_{ij} = E(y_{ij}|\mathbf{x}, \mathbf{u_j}) = \exp(\mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j)$, where $\mathbf{x}_{ij}$ is the $1 \times p$ row vector of the fixed-effects covariates, analogous to the covariates you would find in a standard negative binomial regression model, with regression coefficients (fixed effects) $\boldsymbol{\beta}$; $\mathbf{z}_{ij}$ is the $1 \times q$ vector of the random-effects

covariates 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. One special case 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 $\boldsymbol{\Sigma}$.

Similarly to the standard negative binomial model, we can consider two parameterizations of what we call the conditional overdispersion, the overdispersion conditional on random effects, in a random-effects negative binomial model. For the mean-overdispersion (or, more technically, mean-conditional-overdispersion) parameterization,

$$r_{ij} = 1/\alpha \quad \text{and} \quad p_{ij} = \frac{1}{1+\alpha\mu_{ij}} \tag{2}$$

and the conditional overdispersion is equal to $1 + \alpha\mu_{ij}$. For the constant-overdispersion (or, more technically, constant-conditional-overdispersion) parameterization,

$$r_{ij} = \mu_{ij}/\delta \quad \text{and} \quad p_{ij} = \frac{1}{1+\delta} \tag{3}$$

and the conditional overdispersion is equal to $1 + \delta$. In what follows, for brevity, we will use the term overdispersion parameter to mean conditional overdispersion parameter, unless stated otherwise.

In the context of random-effects negative binomial models, it is important to decide which model is used as a reference model for the definition of the overdispersion. For example, if we consider a corresponding random-effects Poisson model as a comparison model, the parameters α and δ can still be viewed as unconditional overdispersion parameters, as we show below, although the notion of a constant overdispersion is no longer applicable.

If we retain the definition of the overdispersion as the excess variation with respect to a Poisson process for which the variance is equal to the mean, we need to carefully distinguish between the marginal (unconditional) mean with random effects integrated out and the conditional mean given random effects.

In what follows, for simplicity, we omit the dependence of the formulas on $\mathbf{x}$. Contingent on random effects, the (conditional) dispersion $\mathrm{Var}(y_{ij}|\mathbf{u_j}) = (1+\alpha\mu_{ij})\mu_{ij}$ for the mean parameterization and $\mathrm{Var}(y_{ij}|\mathbf{u_j}) = (1+\delta)\mu_{ij}$ for the constant parameterization; the usual interpretation of the parameters holds (conditionally).

If we consider the marginal mean or, specifically, the marginal dispersion for, for example, a two-level random-intercept model, then

$$\mathrm{Var}(y_{ij}) = \left[1 + \{\exp(\sigma^2)(1+\alpha) - 1\}E(y_{ij})\right] E(y_{ij})$$

for the mean parameterization and

$$\mathrm{Var}(y_{ij}) = \left[1 + \delta + \{\exp(\sigma^2) - 1\}E(y_{ij})\right] E(y_{ij})$$

for the constant parameterization, where σ^2 is the variance component corresponding to the random intercept.

A few things of interest compared with the standard negative binomial model. First, the random-effects negative binomial model is not strictly an overdispersed model. The combination of values of α and σ^2 can lead to an underdispersed model, a model with smaller variability than the Poisson variability. Underdispersed models are not as common in practice, so we will concentrate on the overdispersion in this entry. Second, α (or δ) no longer solely determine the overdispersion and thus cannot be viewed as unconditional overdispersion parameters. Overdispersion is now a function of both α (or δ) and σ^2. Third, the notion of a constant overdispersion is not applicable.

Two special cases are worth mentioning. When $\sigma^2 = 0$, the dispersion reduces to that of a standard negative binomial model. When $\alpha = 0$ (or $\delta = 0$), the dispersion reduces to that of a two-level random-intercept Poisson model, which itself is, in general, an overdispersed model; see Rabe-Hesketh and Skrondal (2012, chap. 13.7) for more details. As such, α and δ retain the typical interpretation as dispersion parameters relative to a random-intercept Poisson model.

Below we present two short examples of mixed-effects negative binomial regression; refer to [ME] **me** and [ME] **meglm** for more examples including crossed-effects models.

▷ Example 1: Two-level random-intercept model

Rabe-Hesketh and Skrondal (2012, chap. 13.7) analyze the data from Winkelmann (2004) on the impact of the 1997 health reform in Germany on the number of doctor visits. The intent of policymakers was to reduce government expenditures on healthcare. We use a subsample of the data restricted to 1,158 women who were employed full time the year before or after the reform.

```
. use http://www.stata-press.com/data/r15/drvisits
. describe
Contains data from http://www.stata-press.com/data/r15/drvisits.dta
  obs:         2,227
 vars:             8                          23 Jan 2016 18:39
 size:        71,264
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
id              float   %9.0g                 person id
numvisit        float   %9.0g                 number of doctor visits in the
                                                last 3 months before interview
age             float   %9.0g                 age in years
educ            float   %9.0g                 education in years
married         float   %9.0g                 =1 if married, 0 otherwise
badh            float   %9.0g                 self-reported health status, =1
                                                if bad
loginc          float   %9.0g                 log of household income
reform          float   %9.0g                 =0 if interview before reform, =1
                                                if interview after reform
-------------------------------------------------------------------------------
Sorted by:
```

The dependent variable, `numvisit`, is a count of doctor visits. The covariate of interest is a dummy variable, `reform`, which indicates whether a doctor visit took place before or after the reform. Other covariates include a self-reported health status, age, education, marital status, and a log of household income.

We first fit a two-level random-intercept Poisson model. We specify the random intercept at the `id` level, that is, an individual-person level.

```
. mepoisson numvisit reform age educ married badh loginc || id:, irr

Fitting fixed-effects model:

Iteration 0:   log likelihood = -9326.8542
Iteration 1:   log likelihood = -5989.7308
Iteration 2:   log likelihood = -5942.7581
Iteration 3:   log likelihood = -5942.7243
Iteration 4:   log likelihood = -5942.7243

Refining starting values:

Grid node 0:   log likelihood = -4761.1257

Fitting full model:

Iteration 0:   log likelihood = -4761.1257
Iteration 1:   log likelihood = -4683.2239
Iteration 2:   log likelihood = -4646.9329
Iteration 3:   log likelihood =  -4645.736
Iteration 4:   log likelihood = -4645.7371
Iteration 5:   log likelihood = -4645.7371

Mixed-effects Poisson regression               Number of obs     =      2,227
Group variable:                id              Number of groups  =      1,518

                                               Obs per group:
                                                             min =          1
                                                             avg =        1.5
                                                             max =          2

Integration method: mvaghermite                Integration pts.  =          7

                                               Wald chi2(6)      =     249.37
Log likelihood = -4645.7371                    Prob > chi2       =     0.0000
```

numvisit	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
reform	.9517026	.0309352	-1.52	0.128	.8929617	1.014308
age	1.005821	.002817	2.07	0.038	1.000315	1.011357
educ	1.008788	.0127394	0.69	0.488	.9841258	1.034068
married	1.082078	.0596331	1.43	0.152	.9712905	1.205503
badh	2.471857	.151841	14.73	0.000	2.191471	2.788116
loginc	1.094144	.0743018	1.32	0.185	.9577909	1.249909
_cons	.5216748	.2668604	-1.27	0.203	.191413	1.421766
id						
var(_cons)	.8177932	.0503902			.724761	.9227673

```
Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test vs. Poisson model: chibar2(01) = 2593.97      Prob >= chibar2 = 0.0000

. estimates store mepoisson
```

Because we specified the `irr` option, the parameters are reported as incidence-rate ratios. The healthcare reform seems to reduce the expected number of visits by 5% but without statistical significance.

Because we have only one random effect at the `id` level, the table shows only one variance component. The estimate of σ_u^2 is 0.82 with standard error 0.05. The reported likelihood-ratio test shows that there is enough variability between women to favor a mixed-effects Poisson regression over a standard Poisson regression; see *Distribution theory for likelihood-ratio test* in [ME] **me** for a discussion of likelihood-ratio testing of variance components.

It is possible that after conditioning on the person-level random effect, the counts of doctor visits are overdispersed. For example, medical problems occurring during the time period leading to the survey can result in extra doctor visits. We thus reexamine the data with `menbreg`.

```
. menbreg numvisit reform age educ married badh loginc || id:, irr

Fitting fixed-effects model:

Iteration 0:   log likelihood = -4610.7165
Iteration 1:   log likelihood = -4563.4682
Iteration 2:   log likelihood = -4562.3241
Iteration 3:   log likelihood = -4562.3238

Refining starting values:

Grid node 0:   log likelihood = -4643.5216

Fitting full model:

Iteration 0:   log likelihood = -4643.5216  (not concave)
Iteration 1:   log likelihood =  -4555.961
Iteration 2:   log likelihood = -4518.7353
Iteration 3:   log likelihood = -4513.1951
Iteration 4:   log likelihood = -4513.1853
Iteration 5:   log likelihood = -4513.1853

Mixed-effects nbinomial regression              Number of obs     =      2,227
Overdispersion:            mean
Group variable:              id                 Number of groups  =      1,518

                                                Obs per group:
                                                              min =          1
                                                              avg =        1.5
                                                              max =          2

Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(6)      =     237.35
Log likelihood = -4513.1853                     Prob > chi2       =     0.0000
```

numvisit	IRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
reform	.9008536	.042022	-2.24	0.025	.8221449	.9870975
age	1.003593	.0028206	1.28	0.202	.9980799	1.009137
educ	1.007026	.012827	0.55	0.583	.9821969	1.032483
married	1.089597	.064213	1.46	0.145	.970738	1.223008
badh	3.043562	.2366182	14.32	0.000	2.613404	3.544523
loginc	1.136342	.0867148	1.67	0.094	.9784833	1.319668
_cons	.5017199	.285146	-1.21	0.225	.1646994	1.528377
/lnalpha	-.7962692	.1190614			-1.029625	-.5629132
id						
var(_cons)	.4740088	.0582404			.3725642	.6030754

```
Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline incidence rate (conditional on zero random
      effects).
LR test vs. nbinomial model: chibar2(01) = 98.28      Prob >= chibar2 = 0.0000
```

The estimated effect of the healthcare reform now corresponds to the reduction in the number of doctor visits by 10%—twice as much compared with the Poisson model—and this effect is significant at the 5% level.

The estimate of the variance component σ_u^2 drops down to 0.47 compared with `mepoisson`, which is not surprising given that now we have an additional parameter that controls the variability of the data.

Because the conditional overdispersion α is assumed to be greater than 0, it is parameterized on the log scale, and its log estimate is reported as `/lnalpha` in the output. In our model, $\widehat{\alpha} = \exp(-0.80) = 0.45$. We can also compute the unconditional overdispersion in this model by using $\exp(.47) \times (1 + .45) - 1 = 1.32$.

The reported likelihood-ratio test shows that there is enough variability between women to favor a mixed-effects negative binomial regression over negative binomial regression without random effects.

We can also perform a likelihood-ratio test comparing the mixed-effects negative binomial model to the mixed-effects Poisson model. Because we are comparing two different estimators, we need to use the `force` option with `lrtest`. In general, there is no guarantee as to the validity or interpretability of the resulting likelihood-ratio test, but in our case we know the test is valid because the mixed-effects Poisson model is nested within the mixed-effects negative binomial model.

```
. lrtest mepoisson ., force
Likelihood-ratio test                                 LR chi2(1)  =     265.10
(Assumption: mepoisson nested in .)                   Prob > chi2 =     0.0000
Note: The reported degrees of freedom assumes the null hypothesis is not on
      the boundary of the parameter space.  If this is not true, then the
      reported test is conservative.
```

The reported likelihood-ratio test favors the mixed-effects negative binomial model. The reported test is conservative because the test of $H_0: \alpha = 0$ occurs on the boundary of the parameter space; see *Distribution theory for likelihood-ratio test* in [ME] **me** for details.

◁

The above extends to models with more than two levels of nesting by adding more random-effects equations, each separated by `||`. The order of nesting goes from left to right as the groups go from biggest (highest level) to smallest (lowest level). To demonstrate a three-level model, we revisit example 3 from [ME] **mepoisson**.

▷ Example 2: Three-level random-intercept model

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/r15/melanoma
(Skin cancer (melanoma) data)
. describe
Contains data from http://www.stata-press.com/data/r15/melanoma.dta
  obs:           354                          Skin cancer (melanoma) data
 vars:             6                          30 May 2016 17:10
 size:         4,956                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     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. The variable `uv` is a measure of exposure to ultraviolet (UV) radiation.

In example 3 of [ME] **mepoisson**, we noted that because counties also identified the observations, we could model overdispersion by using a four-level Poisson model with a random intercept at the county level. Here we fit a three-level negative binomial model with the default mean-dispersion parameterization.

```
. menbreg deaths uv, exposure(expected) || nation: || region:
Fitting fixed-effects model:
Iteration 0:   log likelihood =  -1361.855
Iteration 1:   log likelihood = -1230.0211
Iteration 2:   log likelihood =  -1211.049
Iteration 3:   log likelihood = -1202.5641
Iteration 4:   log likelihood = -1202.5329
Iteration 5:   log likelihood = -1202.5329
Refining starting values:
Grid node 0:   log likelihood = -1209.6951
Fitting full model:
Iteration 0:   log likelihood = -1209.6951  (not concave)
  (output omitted)
Iteration 11:  log likelihood = -1086.3902
Mixed-effects nbinomial regression              Number of obs     =        354
Overdispersion:            mean
```

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

```
Integration method: mvaghermite                 Integration pts.  =          7
                                                Wald chi2(1)      =       8.73
Log likelihood = -1086.3902                     Prob > chi2       =     0.0031
```

deaths	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
uv	-.0335933	.0113725	-2.95	0.003	-.055883	-.0113035
_cons	-.0790606	.1295931	-0.61	0.542	-.3330583	.1749372
ln(expected)	1	(exposure)				
/lnalpha	-4.182603	.3415036			-4.851937	-3.513268
nation var(_cons)	.1283614	.0678971			.0455187	.3619758
nation> region var(_cons)	.0401818	.0104855			.0240938	.067012

```
LR test vs. nbinomial model: chi2(2) = 232.29             Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

The estimates are very close to those of mepoisson. The conditional overdispersion in our model is $\widehat{\alpha} = \exp(-4.18) = 0.0153$. It is in agreement with the estimate of the random intercept at the county level, 0.0146, in a four-level random-effects Poisson model reported by mepoisson. Because the negative binomial is a three-level model, we gained some computational efficiency over the four-level Poisson model.

◁

Stored results

menbreg stores the following in e():

Scalars

e(N)	number of observations
e(k)	number of parameters
e(k_dv)	number of dependent variables
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
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(N_clust)	number of clusters
e(chi2)	χ^2
e(p)	p-value for model test
e(ll_c)	log likelihood, comparison model
e(chi2_c)	χ^2, comparison test
e(df_c)	degrees of freedom, comparison test
e(p_c)	p-value for comparison test
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	meglm
e(cmd2)	menbreg
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type
e(wexp)	weight expression (first-level weights)
e(fweight*k*)	fweight variable for *k*th highest level, if specified
e(iweight*k*)	iweight variable for *k*th highest level, if specified
e(pweight*k*)	pweight variable for *k*th highest level, if specified
e(covariates)	list of covariates
e(ivars)	grouping variables
e(model)	nbreg
e(title)	title in estimation output
e(link)	log
e(family)	nbinomial
e(clustvar)	name of cluster variable
e(dispersion)	mean or constant
e(offset)	offset
e(intmethod)	integration method
e(n_quad)	number of integration points
e(chi2type)	Wald; type of model χ^2
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. Err.
e(opt)	type of optimization

e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(user)	name of likelihood-evaluator program
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(marginswtype)	weight type for margins
e(marginswexp)	weight expression 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(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
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
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample

Methods and formulas

menbreg is a convenience command for meglm with a log link and an nbinomial family; see [ME] **meglm**.

Without a loss of generality, consider a two-level negative binomial 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 $\mathbf{u}_j$ and the conditional overdispersion parameter α in a mean-overdispersion parameterization, is

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j,\alpha) &= \prod_{i=1}^{n_j}\left\{\frac{\Gamma(y_{ij}+r)}{\Gamma(y_{ij}+1)\Gamma(r)}p_{ij}^{r}(1-p_{ij})^{y_{ij}}\right\} \\
&= \exp\left[\sum_{i=1}^{n_j}\{\log\Gamma(y_{ij}+r) - \log\Gamma(y_{ij}+1) - \log\Gamma(r) + c(y_{ij},\alpha)\}\right]
\end{aligned}
$$

where $c(y_{ij},\alpha)$ is defined as

$$
-\frac{1}{\alpha}\log\{1+\exp(\eta_{ij}+\log\alpha)\} - y_{ij}\log\{1+\exp(-\eta_{ij}-\log\alpha)\}
$$

and $r = 1/\alpha$, $p_{ij} = 1/(1+\alpha\mu_{ij})$, and $\eta_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta} + \mathbf{z}_{ij}\mathbf{u}_j$.

For the constant-overdispersion parameterization with the conditional overdispersion parameter δ, the conditional distribution of $\mathbf{y}_j$ is

$$
\begin{aligned}
f(\mathbf{y}_j|\mathbf{u}_j,\delta) &= \prod_{i=1}^{n_j}\left\{\frac{\Gamma(y_{ij}+r_{ij})}{\Gamma(y_{ij}+1)\Gamma(r_{ij})}p^{r_{ij}}(1-p)^{y_{ij}}\right\} \\
&= \exp\left[\sum_{i=1}^{n_j}\{\log\Gamma(y_{ij}+r_{ij}) - \log\Gamma(y_{ij}+1) - \log\Gamma(r_{ij}) + c(y_{ij},\delta)\}\right]
\end{aligned}
$$

where $c(y_{ij},\delta)$ is defined as

$$
-\left(\frac{\mu_{ij}}{\delta}+y_{ij}\right)\log(1+\delta) + y_{ij}\log\delta
$$

and $r_{ij} = \mu_{ij}/\delta$ and $p = 1/(1+\delta)$.

For conciseness, let γ denote either conditional overdispersion parameter. 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 $f(\mathbf{y}_j,\mathbf{u}_j,\gamma)$,

$$
\begin{aligned}
\mathcal{L}_j(\boldsymbol{\beta},\boldsymbol{\Sigma},\gamma) &= (2\pi)^{-q/2}\,|\boldsymbol{\Sigma}|^{-1/2}\int f(\mathbf{y}_j|\mathbf{u}_j,\gamma)\exp\left(-\mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2\right)d\mathbf{u}_j \\
&= (2\pi)^{-q/2}\,|\boldsymbol{\Sigma}|^{-1/2}\int \exp\left\{h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j,\gamma\right)\right\}d\mathbf{u}_j
\end{aligned}
\tag{4}
$$

where

$$
h\left(\boldsymbol{\beta},\boldsymbol{\Sigma},\mathbf{u}_j,\gamma\right) = f(\mathbf{y}_j|\mathbf{u}_j,\gamma) - \mathbf{u}_j'\boldsymbol{\Sigma}^{-1}\mathbf{u}_j/2
$$

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 (4) has no closed form and thus must be approximated; see *Methods and formulas* in [ME] **meglm** for details.

`menbreg` supports multilevel weights and survey data; see *Methods and formulas* in [ME] **meglm** for details.

References

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.

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

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

Winkelmann, R. 2004. Health care reform and the number of doctor visits—An econometric analysis. *Journal of Applied Econometrics* 19: 455–472.

Also see

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

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

[ME] **meqrpoisson** — Multilevel mixed-effects Poisson regression (QR decomposition)

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

[BAYES] **bayes: menbreg** — Bayesian multilevel negative binomial regression

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

[SVY] **svy estimation** — Estimation commands for survey data

[XT] **xtnbreg** — Fixed-effects, random-effects, & population-averaged negative binomial models

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