

meqrpoisson postestimation — Postestimation tools for meqrpoisson

[Postestimation commands](#)
[predict](#)
[margins](#)
[Remarks and examples](#)
[Methods and formulas](#)
[References](#)
[Also see](#)

Postestimation commands

The following postestimation commands are of special interest after `meqrpoisson`:

Command	Description
<code>estat group</code>	summarize the composition of the nested groups
<code>estat recovariance</code>	display the estimated random-effects covariance matrices
<code>estat sd</code>	display variance components as standard deviations and correlations

The following standard postestimation commands are also available:

Command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of estimates
<code>estat ic</code>	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
<code>estat summarize</code>	summary statistics for the estimation sample
<code>estat vce</code>	variance–covariance matrix of the estimators (VCE)
<code>estimates</code>	cataloging estimation results
<code>hausman</code>	Hausman's specification test
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of coefficients
<code>lrtest</code>	likelihood-ratio test
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of estimates
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as mean responses; linear predictions; standard errors; and Pearson, deviance, and Anscombe residuals.

Menu for predict

Statistics > Postestimation

Syntax for predict

Syntax for obtaining estimated random effects and their standard errors

```
predict [type] newvarsspec [if] [in], reflects [reses(newvarsspec)
      relevel(levelvar) ]
```

Syntax for obtaining other predictions

```
predict [type] newvar [if] [in] [, statistic nooffset fixedonly]
```

newvarsspec is *stub** or *newvarlist*.

<i>statistic</i>	Description
------------------	-------------

Main	
<code>mu</code>	predicted mean; the default
<code>xb</code>	linear predictor for the fixed portion of the model only
<code>stdp</code>	standard error of the fixed-portion linear prediction
<code>pearson</code>	Pearson residuals
<code>deviance</code>	deviance residuals
<code>anscombe</code>	Anscombe residuals

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`reflects` calculates posterior modal estimates of the random effects. By default, estimates for all random effects in the model are calculated. However, if the `relevel`(*levelvar*) option is specified, then estimates for only level *levelvar* in the model are calculated. For example, if `classes` are nested within `schools`, then typing

```
. predict b*, reflects relevel(school)
```

would yield random-effects estimates at the school level. You must specify *q* new variables, where *q* is the number of random-effects terms in the model (or level). However, it is much easier to just specify *stub** and let Stata name the variables *stub1*, *stub2*, ..., *stubq* for you.

`reses(newvarsspec)` calculates standard errors for the random-effects estimates. By default, standard errors for all random effects in the model are calculated. However, if the `relevel(levelvar)` option is specified, then standard errors for only level *levelvar* in the model are calculated; see the `reffects` option.

You must specify q new variables, where q is the number of random-effects terms in the model (or level). However, it is much easier to just specify `stub*` and let Stata name the variables `stub1`, `stub2`, \dots , `stub q` for you. The new variables will have the same storage type as the corresponding random-effects variables.

The `reffects` and `reses()` options often generate multiple new variables at once. When this occurs, the random effects (or standard errors) contained in the generated variables correspond to the order in which the variance components are listed in the output of `meqrpoisson`. Still, examining the variable labels of the generated variables (with the `describe` command, for instance) can be useful in deciphering which variables correspond to which terms in the model.

`relevel(levelvar)` specifies the level in the model at which predictions for random effects and their standard errors are to be obtained. *levelvar* is the name of the model level and is either the name of the variable describing the grouping at that level or is `_all`, a special designation for a group comprising all the estimation data.

`mu`, the default, calculates the predicted mean, that is, the predicted count. By default, this is based on a linear predictor that includes both the fixed effects and the random effects, and the predicted mean is conditional on the values of the random effects. Use the `fixedonly` option (see [below](#)) if you want predictions that include only the fixed portion of the model, that is, if you want random effects set to 0.

`xb` calculates the linear prediction $\mathbf{x}\beta$ based on the estimated fixed effects (coefficients) in the model. This is equivalent to fixing all random effects in the model to their theoretical (prior) mean value of 0.

`stdp` calculates the standard error of the fixed-effects linear predictor $\mathbf{x}\beta$.

`pearson` calculates Pearson residuals. Pearson residuals large in absolute value may indicate a lack of fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`deviance` calculates deviance residuals. Deviance residuals are recommended by [McCullagh and Nelder \(1989\)](#) as having the best properties for examining the goodness of fit of a GLM. They are approximately normally distributed if the model is correctly specified. They may be plotted against the fitted values or against a covariate to inspect the model's fit. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`anscombe` calculates Anscombe residuals, which are designed to closely follow a normal distribution. By default, residuals include both the fixed portion and the random portion of the model. The `fixedonly` option modifies the calculation to include the fixed portion only.

`nooffset` is relevant only if you specified `offset(varnameo)` or `exposure(varnamee)` for `meqrpoisson`. It modifies the calculations made by `predict` so that they ignore the offset/exposure variable; the linear prediction is treated as $\mathbf{X}\beta + \mathbf{Z}\mathbf{u}$ rather than $\mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \text{offset}$, or $\mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \ln(\text{exposure})$, whichever is relevant.

`fixedonly` modifies predictions to include only the fixed portion of the model, equivalent to setting all random effects equal to 0; see the [mu](#) option.

margins

Description for margins

`margins` estimates margins of response for linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

```
margins [marginlist] [, options]  
margins [marginlist] , predict(statistic ...) [options]
```

<i>statistic</i>	Description
<code>xb</code>	linear predictor for the fixed portion of the model only; the default
<code><u>r</u>effects</code>	not allowed with <code>margins</code>
<code>mu</code>	not allowed with <code>margins</code>
<code>stdp</code>	not allowed with <code>margins</code>
<code><u>p</u>earson</code>	not allowed with <code>margins</code>
<code><u>d</u>eviance</code>	not allowed with <code>margins</code>
<code><u>a</u>nscombe</code>	not allowed with <code>margins</code>

Statistics not allowed with `margins` are functions of stochastic quantities other than $e(b)$.

For the full syntax, see [R] [margins](#).

Remarks and examples

[stata.com](http://www.stata.com)

Various predictions, statistics, and diagnostic measures are available after fitting a Poisson mixed-effects model with `meqrpoisson`. For the most part, calculation centers around obtaining estimates of the subject/group-specific random effects. Random effects are not estimated when the model is fit but instead need to be predicted after estimation.

► Example 1: Obtaining predicted counts and estimates of random effects

In [example 2](#) of [ME] [meqrpoisson](#), we modeled the number of deaths among males in nine European nations as a function of exposure to ultraviolet radiation (`uv`). We used a three-level Poisson model with random effects at the nation and region levels.

```
. use http://www.stata-press.com/data/r15/melanoma  
(Skin cancer (melanoma) data)  
. meqrpoisson deaths c.uv##c.uv, exposure(expected) || nation: || region:  
(output omitted)
```

We can use `predict` to obtain the predicted counts as well as the estimates of the random effects at the nation and region levels.

```
. predict mu
(option mu assumed; predicted means)
. predict re_nat re_reg, reffects
```

Stata displays a note that the predicted values of `mu` are based on the posterior means of random effects. You can use the `modes` option to obtain predictions based on the posterior modes of random effects.

Here we list the data for the first nation in the dataset, which happens to be Belgium:

```
. list nation region deaths mu re_nat re_reg if nation==1, sepy(region)
```

	nation	region	deaths	mu	re_nat	re_reg
1.	Belgium	1	79	69.53809	-.1195986	.3621477
2.	Belgium	2	80	78.28663	-.1195986	.0478318
3.	Belgium	2	51	46.30186	-.1195986	.0478318
4.	Belgium	2	43	54.35935	-.1195986	.0478318
5.	Belgium	2	89	66.90427	-.1195986	.0478318
6.	Belgium	2	19	34.89818	-.1195986	.0478318
7.	Belgium	3	19	8.199643	-.1195986	-.4348477
8.	Belgium	3	15	41.09571	-.1195986	-.4348477
9.	Belgium	3	33	30.90989	-.1195986	-.4348477
10.	Belgium	3	9	6.942485	-.1195986	-.4348477
11.	Belgium	3	12	12.21362	-.1195986	-.4348477

We can see that the predicted random effects at the nation level, `re_nat`, are the same for all the observations listed. Similarly, the predicted random effects at the region level, `re_reg`, are the same within each region.

◀

□ Technical note

Out-of-sample predictions are permitted after `meqrpoisson`, but if these predictions involve estimated random effects, the integrity of the estimation data must be preserved. If the estimation data have changed since the model was fit, `predict` will be unable to obtain predicted random effects that are appropriate for the fitted model and will give an error. Thus to obtain out-of-sample predictions that contain random-effects terms, be sure that the data for these predictions are in observations that augment the estimation data.

□

Methods and formulas

Continuing the discussion in *Methods and formulas* of [ME] `meqrpoisson` and using the definitions and formulas defined there, we begin by considering the prediction of the random effects \mathbf{u}_j for the j th cluster in a two-level model.

Given a set of estimated meqrpoisson parameters, $(\widehat{\beta}, \widehat{\Sigma})$, a profile likelihood in \mathbf{u}_j is derived from the joint distribution $f(\mathbf{y}_j, \mathbf{u}_j)$ as

$$\mathcal{L}_j(\mathbf{u}_j) = \exp\{-c(\mathbf{y}_j)\} (2\pi)^{-q/2} |\widehat{\Sigma}|^{-1/2} \exp\left\{g\left(\widehat{\beta}, \widehat{\Sigma}, \mathbf{u}_j\right)\right\} \quad (1)$$

The conditional maximum likelihood estimator of \mathbf{u}_j —conditional on fixed $(\widehat{\beta}, \widehat{\Sigma})$ —is the maximizer of $\mathcal{L}_j(\mathbf{u}_j)$ or, equivalently, the value of $\widehat{\mathbf{u}}_j$ that solves

$$\mathbf{0} = g'\left(\widehat{\beta}, \widehat{\Sigma}, \widehat{\mathbf{u}}_j\right) = \mathbf{Z}'_j \left\{ \mathbf{y}_j - \mathbf{m}(\widehat{\beta}, \widehat{\mathbf{u}}_j) \right\} - \widehat{\Sigma}^{-1} \widehat{\mathbf{u}}_j$$

Because (1) is proportional to the conditional density $f(\mathbf{u}_j | \mathbf{y}_j)$, you can also refer to $\widehat{\mathbf{u}}_j$ as the conditional mode (or posterior mode if you lean toward Bayesian terminology). Regardless, you are referring to the same estimator.

Conditional standard errors for the estimated random effects are derived from standard theory of maximum likelihood, which dictates that the asymptotic variance matrix of $\widehat{\mathbf{u}}_j$ is the negative inverse of the Hessian, which is estimated as

$$g''\left(\widehat{\beta}, \widehat{\Sigma}, \widehat{\mathbf{u}}_j\right) = - \left\{ \mathbf{Z}'_j \mathbf{V}(\widehat{\beta}, \widehat{\mathbf{u}}_j) \mathbf{Z}_j + \widehat{\Sigma}^{-1} \right\}$$

Similar calculations extend to models with more than one level of random effects; see [Pinheiro and Chao \(2006\)](#).

For any observation i in the j th cluster in a two-level model, define the linear predictor as

$$\widehat{\eta}_{ij} = \mathbf{x}_{ij} \widehat{\beta} + \mathbf{z}_{ij} \widehat{\mathbf{u}}_j$$

In a three-level model, for the i th observation within the j th level-two cluster within the k th level-three cluster,

$$\widehat{\eta}_{ijk} = \mathbf{x}_{ijk} \widehat{\beta} + \mathbf{z}_{ijk}^{(3)} \widehat{\mathbf{u}}_k^{(3)} + \mathbf{z}_{ijk}^{(2)} \widehat{\mathbf{u}}_{jk}^{(2)}$$

where $\mathbf{z}^{(p)}$ and $\mathbf{u}^{(p)}$ refer to the level p design variables and random effects, respectively. For models with more than three levels, the definition of $\widehat{\eta}$ extends in the natural way, with only the notation becoming more complicated.

If the `fixedonly` option is specified, $\widehat{\eta}$ contains the linear predictor for only the fixed portion of the model, for example, in a two-level model $\widehat{\eta}_{ij} = \mathbf{x}_{ij} \widehat{\beta}$. In what follows, we assume a two-level model, with the only necessary modification for multilevel models being the indexing.

The predicted mean conditional on the random effects $\widehat{\mathbf{u}}_j$ is

$$\widehat{\mu}_{ij} = \exp(\widehat{\eta}_{ij})$$

Pearson residuals are calculated as

$$v_{ij}^P = \frac{y_{ij} - \widehat{\mu}_{ij}}{\{V(\widehat{\mu}_{ij})\}^{1/2}}$$

for $V(\widehat{\mu}_{ij}) = \widehat{\mu}_{ij}$.

Deviance residuals are calculated as

$$\nu_{ij}^D = \text{sign}(y_{ij} - \hat{\mu}_{ij}) \sqrt{\hat{d}_{ij}^2}$$

where

$$\hat{d}_{ij}^2 = \begin{cases} 2\hat{\mu}_{ij} & \text{if } y_{ij} = 0 \\ 2 \left\{ y_{ij} \log \left(\frac{y_{ij}}{\hat{\mu}_{ij}} \right) - (y_{ij} - \hat{\mu}_{ij}) \right\} & \text{otherwise} \end{cases}$$

Anscombe residuals are calculated as

$$\nu_{ij}^A = \frac{3 \left(y_{ij}^{2/3} - \hat{\mu}_{ij}^{2/3} \right)}{2\hat{\mu}_{ij}^{1/6}}$$

For a discussion of the general properties of the above residuals, see [Hardin and Hilbe \(2018, chap. 4\)](#).

References

- Hardin, J. W., and J. M. Hilbe. 2018. *Generalized Linear Models and Extensions*. 4th ed. College Station, TX: Stata Press.
- McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall/CRC.
- 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.

Also see

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

[U] [20 Estimation and postestimation commands](#)