

meqrpoisson postestimation — Postestimation tools for meqrpoisson

Description	Syntax for predict	Menu for predict
Options for predict	Syntax for estat	Menu for estat
Options for estat recovariance	Remarks and examples	Stored results
Methods and formulas	References	Also see

Description

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 matrix (or matrices)

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>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

Special-interest postestimation commands

`estat group` reports the number of groups and minimum, average, and maximum group sizes for each level of the model. Model levels are identified by the corresponding group variable in the data. Because groups are treated as nested, the information in this summary may differ from what you would get if you used the `tabulate` command on each group variable individually.

`estat recovariance` displays the estimated variance–covariance matrix of the random effects for each level in the model. Random effects can be either random intercepts, in which case the corresponding rows and columns of the matrix are labeled as `_cons`, or random coefficients, in which case the label is the name of the associated variable in the data.

Syntax for predict

Syntax for obtaining estimated random effects and their standard errors

```
predict [type] { stub* | newvarlist } [if] [in], { reffects | reses }
        [relevel(levelvar)]
```

Syntax for obtaining other predictions

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

<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.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

`reffects` 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*, reffects 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` calculates standard errors for the random-effects estimates obtained by using the `reffects` option. 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. For example, if classes are nested within schools, then typing

```
. predict se*, reses relevel(school)
```

would yield standard errors 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.

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.

Syntax for `estat`

Summarize the composition of the nested groups

```
estat group
```

Display the estimated random-effects covariance matrix (or matrices)

```
estat recovariance [ , relevel(levelvar) correlation matlist_options ]
```

Menu for `estat`

Statistics > Postestimation > Reports and statistics

Options for `estat recovariance`

`relevel`(*levelvar*) specifies the level in the model for which the random-effects covariance matrix is to be displayed and returned in `r(cov)`. By default, the covariance matrices for all levels in the model are displayed. *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.

`correlation` displays the covariance matrix as a correlation matrix and returns the correlation matrix in `r(corr)`.

matlist_options are style and formatting options that control how the matrix (or matrices) is displayed; see [P] [matlist](#) for a list of that are available.

Remarks and examples

[stata.com](https://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

In [example 2](#) of [ME] [meqrpoisson](#), we modeled the number of observed epileptic seizures as a function of treatment with the drug progabide and other covariates,

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

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

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

```

. use http://www.stata-press.com/data/r13/epilepsy
(Epilepsy data; progabide drug treatment)
. 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 = -655.86727
Performing gradient-based optimization:
Iteration 0:   log likelihood = -655.86727
Iteration 1:   log likelihood = -655.6822
Iteration 2:   log likelihood = -655.68103
Iteration 3:   log likelihood = -655.68103
Mixed-effects Poisson regression
Group variable: subject
Number of obs      =      236
Number of groups   =       59
Obs per group: min =        4
                  avg  =       4.0
                  max  =        4
Integration points =    9
Log likelihood = -655.68103
Wald chi2(5)      =     115.56
Prob > chi2       =      0.0000

```

seizures	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
treat	-.9286588	.4021643	-2.31	0.021	-1.716886	-.1404313
lbas	.8849767	.131252	6.74	0.000	.6277275	1.142226
lbas_trt	.3379757	.2044445	1.65	0.098	-.0627281	.7386795
lage	.4767192	.353622	1.35	0.178	-.2163673	1.169806
visit	-.2664098	.1647096	-1.62	0.106	-.5892347	.0564151
_cons	2.099555	.2203712	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 regression: chi2(3) = 324.54 Prob > chi2 = 0.0000

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

The purpose of this model was to allow subject-specific linear log trends over each subject's four doctor visits, after adjusting for the other covariates. The intercepts of these lines are distributed $N(\beta_0, \sigma_u^2)$, and the slopes are distributed $N(\beta_5, \sigma_v^2)$, based on the fixed effects and assumed distribution of the random effects.

We can use `predict` to obtain estimates of the random effects u_j and v_j and combine these with our estimates of β_0 and β_5 to obtain the intercepts and slopes of the linear log trends.

```

. predict re_visit re_cons, reffects
. generate b1 = _b[visit] + re_visit
. generate b0 = _b[_cons] + re_cons
. by subject, sort: generate tolist = _n==1

```

```
. list subject treat b1 b0 if tolist & (subject <=5 | subject >=55)
```

	subject	treat	b1	b0
1.	1	0	-.4284563	2.164691
5.	2	0	-.2727145	2.179111
9.	3	0	.0026486	2.450811
13.	4	0	-.3194157	2.268827
17.	5	0	.6063656	2.123723
217.	55	1	-.2304782	2.311493
221.	56	1	.2904741	3.211369
225.	57	1	-.4831492	1.457485
229.	58	1	-.252236	1.168154
233.	59	1	-.1266651	2.204869

We list these slopes (b1) and intercepts (b0) for five control subjects and five subjects on the treatment.

```
. count if tolist & treat
    31
. count if tolist & treat & b1 < 0
    25
. count if tolist & !treat
    28
. count if tolist & !treat & b1 < 0
    20
```

We also find that 25 of the 31 subjects taking progabide were estimated to have a downward trend in seizures over their four doctor visits, compared with 20 of the 28 control subjects.

We also obtain predictions for number of seizures, and unless we specify the `fixedonly` option, these predictions will incorporate the estimated subject-specific random effects.

```
. predict n
(option mu assumed; predicted means)
. list subject treat visit seizures n if subject <= 2 | subject >= 58, sep(0)
```

	subject	treat	visit	seizures	n
1.	1	0	-.3	5	3.887582
2.	1	0	-.1	3	3.568324
3.	1	0	.1	3	3.275285
4.	1	0	.3	3	3.00631
5.	2	0	-.3	3	3.705628
6.	2	0	-.1	5	3.508926
7.	2	0	.1	3	3.322664
8.	2	0	.3	3	3.14629
229.	58	1	-.3	0	.9972093
230.	58	1	-.1	0	.9481507
231.	58	1	.1	0	.9015056
232.	58	1	.3	0	.8571553
233.	59	1	-.3	1	2.487858
234.	59	1	-.1	4	2.425625
235.	59	1	.1	3	2.364948
236.	59	1	.3	2	2.305789

□ 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. □

Stored results

`estat recovariance` stores the following in `r()`:

Scalars

`r(relevels)` number of levels

Matrices

`r(Cov#)` level-# random-effects covariance matrix

`r(Corr#)` level-# random-effects correlation matrix (if option `correlation` was specified)

For a G -level nested model, # can be any integer between 2 and G .

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

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

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

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

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

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

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

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

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 $\hat{\eta}$ extends in the natural way, with only the notation becoming more complicated.

If the `fixedonly` option is specified, $\hat{\eta}$ contains the linear predictor for only the fixed portion of the model, for example, in a two-level model $\hat{\eta}_{ij} = \mathbf{x}_{ij}\hat{\boldsymbol{\beta}}$. 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 $\hat{\mathbf{u}}_j$ is

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

Pearson residuals are calculated as

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

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

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 \(2012, chap. 4\)](#).

References

- Hardin, J. W., and J. M. Hilbe. 2012. *Generalized Linear Models and Extensions*. 3rd 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](#)