

**fmm postestimation** — Postestimation tools for fmm

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

## Postestimation commands

The following postestimation commands are of special interest after estimation with `fmm`:

Command	Description
<code>estat eform</code>	display exponentiated parameters
<code>estat lcmean</code>	latent class marginal means
<code>estat lcprob</code>	latent class marginal probabilities

The following standard postestimation commands are also available:

Command	Description
<code>contrast</code>	contrasts and linear hypothesis tests
<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>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
* <code>hausman</code>	Hausman's specification test
<code>lincom</code>	linear combination of parameters
* <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

\* `hausman` and `lrtest` are not appropriate with `svy` estimation results.

Postestimation commands such `lincom` and `nlcom` require referencing estimated parameter values, which are accessible via `_b[name]`. To find out what the names are, type `fmm, coeflegend`.

# predict

## Description for predict

`predict` after `fmm` creates new variables containing predictions such as means, probabilities, linear predictions, densities, or latent class probabilities.

## Menu for predict

Statistics > Postestimation

## Syntax for predict

```
predict [type] { stub* | newvarlist } [if] [in] [, statistic options]
```

<i>statistic</i>	Description
Main	
<code>mu</code>	expected value of <i>depvar</i> ; the default
<code>eta</code>	linear prediction of <i>depvar</i>
<code>density</code>	density function at <i>depvar</i>
<code>distribution</code>	distribution function at <i>depvar</i>
<code>survival</code>	survivor function at <i>depvar</i>
<code>classpr</code>	latent class probability
<code>classposteriorpr</code>	posterior latent class probability
<code>score</code>	first derivative of the log likelihood with respect to the parameters

<i>options</i>	Description
Main	
<code>margin</code>	compute <i>statistic</i> marginally with respect to the latent classes
<code>pmargin</code>	compute mu marginally with respect to the posterior latent class probabilities
<code>nooffset</code>	make calculation ignoring offset or exposure
<code>*outcome(<i>depvar</i> [#])</code>	specify observed response variable (default all)
<code>class(#)</code>	specify latent class (default all)

\*`outcome(depvar #)` is allowed only if *depvar* is from `mlogit`, `ologit`, or `oprobit`.  
`outcome(depvar #)` may also be specified as `outcome(#.depvar)` or `outcome(depvar ##)`.  
`outcome(depvar #3)` means the third outcome value. `outcome(depvar #3)` would mean the same as `outcome(depvar 4)` if outcomes were 1, 3, and 4.

## Options for predict

Main

`mu`, the default, calculates the expected value of the outcomes.

`eta` calculates the fitted linear prediction.

**density** calculates the density function. This prediction is computed using the current values of the observed variables, including the dependent variable.

**distribution** calculates the distribution function. This prediction is computed using the current values of the observed variables, including the dependent variable. This option is not allowed for **mlogit** outcomes.

**survival** calculates the survivor function. This prediction is computed using the current values of the observed variables, including the dependent variable. This option is allowed only for **streg** outcomes.

**classpr** calculates predicted probabilities for each latent class.

**classposteriorpr** calculates predicted posterior probabilities for each latent class. The posterior probabilities are a function of the latent-class predictors and the fitted outcome densities.

**scores** calculates the scores for each coefficient in  $e(b)$ . This option requires a new variable list of length equal to the number of columns in  $e(b)$ . Otherwise, use *stub\** to have **predict** generate enumerated variables with prefix *stub*.

**marginal** specifies that the prediction be computed marginally with respect to the latent classes. The marginal prediction is computed by combining the class specific predictions using the latent-class probabilities.

This option is allowed only with **mu** and **density**.

**pmarginal** specifies that the prediction is computed by combining the class specific expected values using the posterior latent-class probabilities.

This option is allowed only with **mu**.

**nooffset** is relevant only if option **offset()** or **exposure()** was specified at estimation time. **nooffset** specifies that **offset()** or **exposure()** be ignored, which produces predictions as if all subjects had equal exposure.

**outcome(*depvar* [#])** specifies the *depvar* for which predictions should be calculated. Predictions for all observed response variables are computed by default. Most models have only one *depvar*. If *depvar* is an **mlogit**, **ologit**, or **oprobit** outcome, then **#** optionally specifies which outcome level to predict. The default is the first level.

**class(#)** specifies that predictions for latent class **#** be calculated. Predictions for all latent classes are computed by default.

# margins

## Description for margins

`margins` estimates margins of response for outcome means, outcome probabilities, and latent-class probabilities.

## Menu for margins

Statistics > Postestimation

## Syntax for margins

```
margins [marginlist] [, options]
```

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

<i>statistic</i>	Description
default	calculate expected values for each <i>depvar</i>
mu	calculate expected value of <i>depvar</i>
eta	calculate expected value of linear prediction of <i>depvar</i>
classpr	calculate latent class prior probabilities
<u>density</u>	not allowed with margins
<u>distribution</u>	not allowed with margins
<u>survival</u>	not allowed with margins
<u>classposteriorpr</u>	not allowed with margins
<u>score</u>	not allowed with margins

`mu` defaults to the first *depvar* if option `outcome()` is not specified. If *depvar* is `mlogit`, `ologit`, or `oprobit`, the default is the first level of the outcome. The default is the first latent class if `class()` is not specified.

`eta` defaults to the first *depvar* if option `outcome()` is not specified. If *depvar* is `mlogit`, the default is the first level of the outcome.

`classpr` defaults to the first latent class if option `class()` is not specified.

`predict`'s option `marginal` is assumed if `predict`'s option `class()` is not specified.

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)

For examples using `estimates stats` to compare models based on Akaike information criterion and Bayesian information criterion, see [FMM] [example 1a](#), [FMM] [example 1b](#), and [FMM] [example 1d](#).

For examples using `estat lcp` to obtain marginal latent class probabilities and `estat lmean` to obtain marginal predicted means, see [FMM] [example 2](#) and [FMM] [example 3](#).

For examples using `test` and `contrast` to test equality of coefficients across classes, see [FMM] [example 1c](#).

For examples using `predict`, see [FMM] [example 2](#), [FMM] [example 3](#), and [FMM] [example 4](#).

## Methods and formulas

See *Methods and formulas* in [FMM] [fmm](#).

## Also see

[FMM] [fmm intro](#) — Introduction to finite mixture models

[FMM] [fmm estimation](#) — Fitting finite mixture models

[FMM] [fmm](#) — Finite mixture models using the fmm prefix