

**predict after gsem** — Generalized linear predictions, etc.

[Description](#)                      [Menu](#)                      [Syntax](#)                      [Options](#)  
[Remarks and examples](#)      [Reference](#)                [Also see](#)

## Description

`predict` is a standard postestimation command of Stata. This entry concerns use of `predict` after `gsem`. See [\[SEM\] predict after sem](#) if you fit your model with `sem`.

`predict` after `gsem` creates new variables containing observation-by-observation values of estimated observed response variables, linear predictions of observed response variables, or endogenous or exogenous latent variables.

## Menu

Statistics > SEM (structural equation modeling) > Predictions

## Syntax

*Syntax for predicting observed endogenous outcomes and other statistics*

```
predict [type] newvarsspec [if] [in] [, statistic options]
```

*Syntax for obtaining estimated latent variables and their standard errors*

```
predict [type] newvarsspec [if] [in], lstatistic [loptions]
```

*Syntax for obtaining ML scores*

```
predict [type] newvarsspec [if] [in], scores
```

*newvarsspec* is *stub\** or *newvarlist*.

The default is to predict observed endogenous variables with empirical Bayes means predictions of the latent variables.

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

---

Main	
<code>mu</code>	expected value of <i>depvar</i> ; the default
<code>pr</code>	probability (synonym for <code>mu</code> when $\mu$ is a probability)
<code>eta</code>	expected value of 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>expression(<i>exp</i>)</code>	calculate prediction using <i>exp</i>

---

## 2 predict after gsem — Generalized linear predictions, etc.

---

*options* Description

---

Main

conditional(*ctype*) compute *statistic* conditional on estimated latent variables; default is `conditional(ebmeans)`

marginal compute *statistic* marginally with respect to the latent variables

nooffset make calculation ignoring offset or exposure

† outcome(*depvar* [#]) specify observed response variable (default all)

Integration

int\_options integration options

---

† `outcome(depvar #)` is allowed only if *depvar* has family `multinomial`, `ordinal`, or `bernoulli`. Predicting other generalized responses requires specifying only `outcome(depvar)`.

`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.

*ctype* Description

---

ebmeans empirical Bayes means of latent variables; the default

ebmodes empirical Bayes modes of latent variables

fixedonly prediction for the fixed portion of the model only

---

*lstatistic* Description

---

Main

latent empirical Bayes prediction of all latent variables

latent(*varlist*) empirical Bayes prediction of specified latent variables

---

*loptions* Description

---

Main

ebmeans empirical Bayes means of latent variables; the default

ebmodes empirical Bayes modes of latent variables

se(*stub\** | *newvarlist*) standard errors of empirical Bayes estimates

Integration

int\_options integration options

---

*int\_options* Description

---

intpoints(#) use # quadrature points to compute marginal predictions and empirical Bayes means

iterate(#) set maximum number of iterations in computing statistics involving empirical Bayes estimators

tolerance(#) set convergence tolerance for computing statistics involving empirical Bayes estimators

---

## Options

Main

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

`pr` calculates predicted probabilities and is a synonym for `mu`. This option is available only for multinomial, ordinal, and Bernoulli 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 multinomial 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 only allowed for exponential, gamma, loglogistic, lognormal, and Weibull outcomes.

`expression(exp)` specifies the prediction as an expression. *exp* is any valid Stata expression, but the expression must contain a call to one of the two special functions unique to this option:

1. `mu(outcome)`: The `mu()` function specifies the calculation of the mean prediction for *outcome*. If `mu()` is specified without *outcome*, the mean prediction for the first outcome is implied.  
`pr(outcome)`: The `pr()` function is a synonym for `mu(outcome)` when *outcome* identifies a multinomial, ordinal, or Bernoulli outcome.
2. `eta(outcome)`: The `eta()` function specifies the calculation of the linear prediction for *outcome*. If `eta()` is specified without *outcome*, the linear predictor for the first outcome is implied.

When you specify *exp*, both of these functions may be used repeatedly, in combination, and in combination with other Stata functions and expressions.

`conditional(ctype)` and `marginal` specify how latent variables are handled in computing *statistic*.

`conditional()` specifies that *statistic* will be computed conditional on specified or estimated latent variables.

`conditional(ebmeans)`, the default, specifies that empirical Bayes means be used as the estimates of the latent variables. These estimates are also known as posterior mean estimates of the latent variables.

`conditional(ebmodes)` specifies that empirical Bayes modes be used as the estimates of the latent variables. These estimates are also known as posterior mode estimates of the latent variables.

`conditional(fixedonly)` specifies that all latent variables be set to zero, equivalent to using only the fixed portion of the model.

`marginal` specifies that the predicted *statistic* be computed marginally with respect to the latent variables, which means that *statistic* is calculated by integrating the prediction function with respect to all the latent variables over their entire support.

Although this is not the default, marginal predictions are often very useful in applied analysis. They produce what are commonly called population-averaged estimates. They are also required by [margins](#).

`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 that predictions for `depvar` be calculated. Predictions for all observed response variables are computed by default. If `depvar` is a multinomial or an ordinal outcome, then `#` optionally specifies which outcome level to predict.

`latent` and `latent(varlist)` specify that the latent variables be estimated using empirical Bayes predictions. By default or if the `ebmeans` option is specified, empirical Bayes means are computed. With the `ebmodes` option, empirical Bayes modes are computed.

`latent` requests empirical Bayes estimates for all latent variables.

`latent(varlist)` requests empirical Bayes estimates for the specified latent variables.

`ebmeans` specifies that empirical Bayes means be used to predict the latent variables.

`ebmodes` specifies that empirical Bayes modes be used to predict the latent variables.

`se(stub* | newvarlist)` calculates standard errors of the empirical Bayes estimators and stores the result in `newvarlist`. This option requires the `latent` or `latent()` option.

`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`.

---

#### Integration

`intpoints(#)` specifies the number of quadrature points used to compute marginal predictions and the empirical Bayes means; the default is the value from estimation.

`iterate(#)` specifies the maximum number of iterations when computing statistics involving empirical Bayes estimators; the default is the value from estimation.

`tolerance(#)` specifies convergence tolerance when computing statistics involving empirical Bayes estimators; the default is the value from estimation.

## Remarks and examples

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

Out-of-sample prediction is allowed

1. if the prediction does not involve latent variables;
2. if the prediction involves latent variables, directly or indirectly, and the option `conditional(fixedonly)` is specified;
3. if the prediction involves latent variables, directly or indirectly, the model is multilevel, and no observational-level latent variables are involved; or
4. if the prediction is marginal with respect to the latent variables.

`predict` has two ways of specifying the names of the variables to be created:

```
. predict stub*, ...
```

or

```
. predict firstname secondname ..., ...
```

The first creates variables named *stub1*, *stub2*, . . . . The second creates variables with names that you specify. We strongly recommend using the *stub\** syntax when creating multiple variables because you have no way of knowing the order in which to specify the individual variable names to correspond to the order in which `predict` will make the calculations. If you use *stub\**, the variables will be labeled and you can rename them.

The second syntax is useful when you create one variable and specify `outcome()`, `expression()`, or `latent()`.

See [\[SEM\] intro 7](#), [\[SEM\] example 28g](#), and [\[SEM\] example 29g](#).

## Reference

Skrondal, A., and S. Rabe-Hesketh. 2009. Prediction in multilevel generalized linear models. *Journal of the Royal Statistical Society, Series A* 172: 659–687.

## Also see

[\[SEM\] gsem](#) — Generalized structural equation model estimation command

[\[SEM\] gsem postestimation](#) — Postestimation tools for `gsem`

[\[SEM\] intro 7](#) — Postestimation tests and predictions

[\[SEM\] example 28g](#) — One-parameter logistic IRT (Rasch) model

[\[SEM\] example 29g](#) — Two-parameter logistic IRT model

[\[SEM\] methods and formulas for gsem](#) — Methods and formulas for `gsem`