Description Remarks and examples Reference Also see

Description

After fitting a model with sem or gsem, you can perform statistical tests, obtain predicted values, and more. Everything you can do is listed below.

sem and gsem vary in the tests and features available after estimation, and we mark whether each test and feature is available after sem, gsem, or both.

Remarks and examples

Remarks are presented under the following headings:

Replaying the model (sem and gsem)

After estimation, you can type sem or gsem without arguments to display the estimation output:

```
. sem
```
(original output reappears)

If you wish to see results in the Bentler–Weeks formulation, after sem estimation type

```
. estat framework
```
(*output omitted*)

See [SEM] **Example 11**.

In many of the postestimation commands listed below for use after sem and gsem, you will need to refer symbolically to particular coefficients. For instance, in the model

```
. sem ... (Y<-x1) ..., ... cov(e.Y1*e.Y2)
```

the symbolic name of the coefficient corresponding to the path Y<-x1 is _b[Y1:x1], and the symbolic name of the coefficient corresponding to the covariance of e.Y1 and e.Y2 is _b[/cov(e.Y1,e.Y2)].

Figuring out what the names are can be difficult, so instead, type

```
. sem, coeflegend
```

or

```
. gsem, coeflegend
```

With this command, sem (gsem) will produce a table looking very much like the estimation output that lists the _b[] notation for the estimated parameters in the model; see [SEM] **Example 8**.

Displaying odds ratios, incidence-rate ratios, etc. (gsem only)

In some generalized linear response functions, exponentiated coefficients have a special meaning. Those special meanings are

Common name	Family	Link	Meaning of exp(coef)
logit	Bernoulli	logit	odds ratio
ologit	ordinal	logit	odds ratio
mlogit	multinomial	logit	relative-risk ratio
Poisson	Poisson	log	incidence-rate ratio
nbreg	nbreg	log	incidence-rate ratio

For survival models, the special meanings of the exponentiated coefficients are

Survival distribution	Meaning of exp(coef)
exponential	hazard ratio
Weibull	hazard ratio
gamma	time ratio
loglogistic	time ratio
lognormal	time ratio

gsem reports coefficients, not exponentiated coefficients. You can obtain exponentiated coefficients and their standard errors by using estat eform after estimation. Using estat eform is no different from redisplaying results. The syntax is

estat eform *equationname*

After gsem, equations are named after the dependent variable, so if you want to see the equation for cases in exponentiated form, you can type estat eform cases.

See [SEM] **Example 33g**, [SEM] **Example 34g**, [SEM] **Example 47g**, [SEM] **Example 48g**, and [SEM] **estat eform**.

Obtaining goodness-of-fit statistics (sem and gsem)

One goodness-of-fit statistic and test is reported at the bottom of the sem output:

```
. sem                  // redisplay results
  (output omitted)
Structural equation model
  (coefficient table omitted)
LR test of model vs. saturated: chi2(2) = 1.78            Prob > chi2 = 0.4111
```

Be warned that this test is based on the assumption of joint normality of the observed variables. In the case of nonnormal data, you may specify the vce(sbentler) option to obtain the Satorra–Bentler (1994) scaled χ^2 test. In either case, the test is a goodness-of-fit test in badness-of-fit units; a significant result implies that the model does not fit well. More mathematically, the null hypothesis of this test is that the fitted covariance matrix and mean vector of the observed variables are equal to the matrix and vector observed in the population as measured by the sample. Remember, however, the goal is not to maximize the goodness of fit. One must not add paths or covariances that are not theoretically meaningful.

In addition, other goodness-of-fit statistics are available:

1. (sem only.) Command estat gof reports a variety of goodness-of-fit statistics; see [SEM] **estat gof** and [SEM] **Example 4**.
2. (sem only.) Command estat eqgof reports R^2-like goodness-of-fit statistics for each equation separately; see [SEM] **estat eqgof** and [SEM] **Example 3**.
3. (sem only.) Command estat ggof reports goodness-of-fit statistics by group when you have estimated using sem's group() option; see [SEM] **estat ggof** and [SEM] **Example 21**.
4. (sem only.) Command estat residuals reports the element-by-element differences between the observed and fitted covariance matrix, and the observed and fitted mean vector, optionally in standardized or in normalized units; see [SEM] **estat residuals** and [SEM] **Example 10**.
5. (sem and gsem.) Commands estat ic and estimates stats report the Akaike and Bayesian information criteria; see [SEM] **Example 51g**, [SEM] **Example 52g**, [R] **estat ic**, and [R] **estimates stats**.
6. (gsem only.) Command estat lcgof reports goodness-of-fit statistics for latent class models; see [SEM] **estat lcgof** and [SEM] **Example 51g**.

Performing tests for including omitted paths and relaxing constraints (sem only)

1. (sem only.) Command estat mindices reports χ^2 modification indices and significance values for each omitted path in the model, along with the expected parameter change; see [SEM] **estat mindices**, [SEM] **Example 5**, and [SEM] **Example 9**.
2. (sem only.) Command estat scoretests performs score tests on each of the linear constraints placed on the paths and covariances; see [SEM] **estat scoretests** and [SEM] **Example 8**.
3. (sem only.) Command estat ginvariant is for use when you have estimated using sem's group() option; see [SEM] **Intro 6**. This command tests whether you can relax constraints that parameters are equal across groups; see [SEM] **estat ginvariant** and [SEM] **Example 22**.

Performing tests of model simplification (sem and gsem)

1. (sem and gsem.) Command test reports Wald tests of single or multiple linear constraints. See [SEM] **Example 8** and [SEM] **test**.
2. (sem and gsem.) Command lrtest reports likelihood-ratio tests of single or multiple linear constraints. See [SEM] **Example 10**, [SEM] **lrtest**, and [SEM] **Example 39g**.
3. (sem only.) Command estat eqtest reports an overall Wald test for each equation in the model, the test corresponding to all coefficients in the equation except the intercept being simultaneously 0; see [SEM] **estat eqtest** and [SEM] **Example 13**.
4. (sem only.) Command estat ginvariant is for use when you have estimated using sem's group() option; see [SEM] **Intro 6**. This command tests whether parameters allowed to vary across groups could be constrained; see [SEM] **estat ginvariant** and [SEM] **Example 22**.
5. (gsem only.) Command lcstats reports the Vuong–Lo–Mendell–Rubin likelihood-ratio test and the Lo–Mendell–Rubin-adjusted likelihood-ratio test, which are commonly used to test for the appropriate number of classes when performing latent class analysis; see [SEM] **lcstats**, [SEM] **Example 51g**, and [SEM] **Example 52g**.

Displaying other results, statistics, and tests (sem and gsem)

1. (sem only.) The estat stdize: command prefix—used in front of test, testnl, lincom, and nlcom—allows you to perform tests on standardized coefficients. See [SEM] **estat stdize** and [SEM] **Example 16**.
2. (sem only.) Command estat teffects reports total effects of one variable on another and decomposes the total effect into direct and indirect effects. Results can be reported in standardized or unstandardized form. See [SEM] **estat teffects** and [SEM] **Example 7**.
3. (sem only.) Command estat stable assesses the stability of nonrecursive structural equation systems; see [SEM] **estat stable** and [SEM] **Example 7**.
4. (sem and gsem.) Command estat summarize reports summary statistics for the observed variables used in the model; see [SEM] **estat summarize**.
5. (sem and gsem.) Command lincom reports the value, standard error, significance, and confidence interval for linear combinations of estimated parameters; see [SEM] **lincom**.
6. (sem and gsem.) Command nlcom reports the value, standard error, significance, and confidence interval for nonlinear (and linear) combinations of estimated parameters; see [SEM] **nlcom** and [SEM] **Example 42g**.
7. (sem and gsem.) Command estat vce reports the variance–covariance matrix of the estimated parameters; see [R] **estat vce**.
8. (gsem only.) Command estat lcmean reports marginal predicted means of each outcome within each latent classes; see [SEM] **estat lcmean**, [SEM] **Example 50g**, and [SEM] **Example 53g**.
9. (gsem only.) Command estat lcprob reports marginal predicted latent class probabilities; see [SEM] **estat lcprob**, [SEM] **Example 50g**, and [SEM] **Example 53g**.

Obtaining predicted values (sem)

You obtain predicted values with the predict command. Below we will write that predictions are the expected values, but be aware that when there are latent variables in your model, predictions are based on predicted scores; the scores can be inconsistent, and thus any prediction based on them can be inconsistent.

Available are the following:

1. predict *newvar*, xb(*odepvarname*) creates new variable *newvar* containing the predicted values for observed endogenous variable *odepvarname*.

   predict *stub**, xb creates new variables *stub*1, *stub*2, … containing the predicted values for all the observed endogenous variables in the model.

   These predicted values are the expected value of the xb given the values of the observed exogenous variables.

2. predict *newvar*, latent(*Lname*) creates new variable *newvar* containing the predicted values of the latent variable *Lname*, whether endogenous or exogenous.

   predict *stub**, latent creates new variables *stub*1, *stub*2, … containing the predicted values for all the latent variables in the model.

Predicted values of latent variables, also known as predicted factor scores, are the expected values of the variables given the values of the observed variables.

3. `predict` *newvar*`, xblatent(`*Lname*`)` creates new variable *newvar* containing the predicted values for latent endogenous variable *Lname*.

   `predict` *stub*`*, xblatent` creates new variables *stub*1, *stub*2, ... containing the predicted values for all the latent endogenous variables in the model.

   `predict` with `xblatent(`*Lname*`)` differs from `latent(`*Lname*`)` in that the factor scores predicted by `latent()` are then used with the linear equation for *Lname* to make the prediction.

4. `predict` *stub*`*, scores` will create a slew of variables, one for each estimated parameter, containing the observation-by-observation values of the first derivative, also known as scores. This command is intended for use by programmers and may only be used after estimation using `method(ml)` or `method(mlmv)`.

See [SEM] **Example 14** and [SEM] **predict after sem**.

Obtaining predicted values (gsem)

`predict` after `gsem` is more complicated than `predict` after `sem` because generalized SEMs are more complicated. Three new issues arise: the prediction of generalized linear response variables as opposed to linear response variables, the optional presence of multilevel continuous latent variables in the model, and the optional presence of categorical latent variables in the model.

Let's start with the response functions of the observed endogenous variables. Corresponding to `predict, xb` after `sem` are the following:

1. `predict` *newvar*`, mu outcome(`*depvar*`)` predicts mean responses in natural units—linear, probability, counts, and so on. Think $g^{-1}(x_i b)$ in the generalized linear model notation.

   With continuous latent variables, calculations are made using the empirical Bayes mean predictions of latent variables, which can be computationally intensive. You can specify that empirical Bayes modes be used by specifying the `conditional(ebmodes)` option. You can also specify that all continuous latent variables be set to their mean value, typically 0, by specifying the `conditional(fixedonly)` option.

   With categorical latent variables, calculations can be made for a specific latent class by specifying the `class(`*lcspec*`)` option.

2. `predict` *newvar*`, mu outcome(`*depvar*`) marginal` also predicts mean responses in natural units, and again it is appropriate to think $g^{-1}(x_i b)$.

   With continuous latent variables, calculations are made by integrating the prediction formula with respect to the continuous latent variables, which can be computationally intensive.

   With categorical latent variables, the overall mean is computed by a weighted sum of the latent class means, where the weights are latent class probabilities.

3. `predict` *newvar*`, mu outcome(`*depvar*`) pmarginal` also predicts mean responses in natural units, and again it is appropriate to think $g^{-1}(x_i b)$.

   The `pmarginal` option is allowed only for models with categorical latent variables. The overall mean is computed by a weighted sum of the latent class means, where the weights are posterior latent class probabilities.

4. The pr option is a synonym for mu when using family-and-link combinations that produce probabilities, such as logit, probit, etc.

5. predict *newvar*, eta outcome(*depvar*) calculates the linear prediction $x_i b$.

   With continuous latent variables, calculations are made using the empirical Bayes mean predictions of latent variables, which can be computationally intensive. You can specify that empirical Bayes modes be used by specifying the conditional(ebmodes) option. You can also specify that all continuous latent variables be set to their mean value, typically 0, by specifying option conditional(fixedonly).

   With categorical latent variables, calculations can be made for a specific latent class by specifying the class(*lcspec*) option.

6. predict *newvar*, density outcome(*depvar*) calculates the density function for *depvar* using the current data and model parameters.

   With continuous latent variables, calculations are made using the empirical Bayes mean predictions of latent variables, which can be computationally intensive. You can change the treatment of latent variables by specifying the conditional(ebmodes), conditional(fixedonly), or marginal option.

   With categorical latent variables, calculations can be made for a specific latent class by specifying the class(*lcspec*) option.

7. predict *newvar*, distribution outcome(*depvar*) calculates the cumulative distribution function for *depvar* using the current data and model parameters.

   With continuous latent variables, calculations are made using the empirical Bayes mean predictions of latent variables, which can be computationally intensive. You can change the treatment of latent variables by specifying the conditional(ebmodes), conditional(fixedonly), or marginal option.

   With categorical latent variables, calculations can be made for a specific latent class by specifying the class(*lcspec*) option.

   The distribution option is not allowed with family(multinomial).

8. predict *newvar*, survival outcome(*depvar*) calculates the cumulative survivor function for *depvar* using the current data and model parameters.

   With continuous latent variables, calculations are made using the empirical Bayes mean predictions of latent variables, which can be computationally intensive. You can change the treatment of latent variables by specifying the conditional(ebmodes), conditional(fixedonly), or marginal option.

   With categorical latent variables, calculations can be made for a specific latent class by specifying the class(*lcspec*) option.

   The survival option is allowed only with family(exponential), family(gamma), family(loglogistic), family(lognormal), and family(weibull).

9. The `outcome()` option varies to accommodate the multiple predictions produced by multinomial logit, ordinal probit, ordinal logit, and ordinal complementary log–log.

    a. `outcome(`*depvar* #`)` specifies that you want the prediction for *depvar* = #.

    b. `outcome(`#`.`*depvar*`)` is another way of specifying that you want the prediction for *depvar* = #.

    c. `outcome(`*depvar* `#`#`)` specifies that you want the prediction for the #th category.

    For multinomial logit, you use the above form of `outcome()` when requesting `mu` or `eta` predictions.

    For ordered outcomes, you use the above form of `outcome()` when requesting `mu` predictions. For `eta` predictions, there is only one linear equation, and you use the usual `outcome(`*depvar*`)` form.

10. The `class(`*lclspec*`)` option is for models with categorical latent variables and allows you to target predictions for a specific latent class. For models with one categorical latent variable, *lclspec* can be a level value, such as `class(2)` or its equivalent factor-variable notation `class(2.C)`, assuming the latent class variable is `C`. For models with two or more categorical latent variables, *lclspec* may only be in factor-variable notation, such as `class(2.C#1.D)` for categorical latent variables `C` and `D`.

11. Some GLM families, such as Poisson, allow specification of an offset or exposure. If you use one of those families and you specify an offset or exposure with `gsem` when you fit the model and now want predictions with offset = 0 (exposure = 1), you can specify the `nooffset` option.

12. To obtain predicted values for all observed endogenous variables, you can omit the `outcome()` option and replace *newvar* with *stub*`*`—that is, a piece of a new variable name followed by `*`. This is not just convenient; it can also save computer time because some parts of the calculation can be shared.

    For example, you can change

    ```
    predict newvar, mu outcome(depvar)
    ```

    to be

    ```
    predict new*, mu
    ```

So much for the observed endogenous variables. Let's now move on to continuous latent variables, whether endogenous or exogenous:

13. `predict` *newvar*`, latent(`*Lname*`)` predicts the value of the specified latent variable by using empirical Bayes means. You can specify that empirical Bayes modes be used by specifying the `ebmodes` option; see item 16b below.

14. `predict` *newvar*`, latent(`*Lname*`) se(`*another_newvar*`)` will predict both the latent variable and its standard error.

15. You can use the same *stub*`*` trick mentioned for predicting observed endogenous variables to obtain all the latent variables. Replace *newvar* with *stub*`*` and omit *Lname* to produce

    ```
    predict stub*, latent
    ```

    You can do this with the `se()` option, too:

    ```
    predict stub*, latent se(another_stub*)
    ```

16. Four options are related to the iterative procedure for obtaining the predicted values:

    a. `ebmodes` specifies that empirical Bayes modes rather than means be predicted. Because latent variables are assumed to be normally distributed, means and modes are usually similar. They are, however, calculated differently. The predictions are the means or modes of the empirical posterior distributions of the latent variables, which are not necessarily normal. Therefore, the means can differ from the modes. Both are computationally intensive.

       The default empirical Bayes means calculation uses adaptive quadrature, which is to say, numerical integration. All three of the options listed below apply to this calculation.

       The alternative empirical Bayes mode calculation uses an optimization method that does not require adaptive quadrature. For small datasets, modes can usually be calculated more quickly than means. For large datasets, however, predicting modes can actually take longer. If you specify `ebmodes`, of the three options listed below, only `tolerance()` and `iterate()` are relevant.

    b. `intpoints(`#`)` specifies the number of quadrature (numerical integration) points to be used to compute the empirical Bayes means. The default is the value used in estimation, which in turn defaults to 7.

       This option also controls the number of quadrature points used to compute marginal predictions for the observed endogenous variables.

    c. `tolerance(`#`)` specifies the convergence tolerance in computing empirical Bayes means and modes and defaults to the value used in estimation, which in turn defaults to 1.0e–8.

    d. `iterate(`#`)` specifies the maximum number of iterations in computing empirical Bayes means and modes and defaults to the value used in estimation, which in turn defaults to 1,001.

Continuous latent variables come in observation-level and multilevel flavors. If you have a multilevel latent variable such as `M1[school]`, you can predict it just as you would any other latent variable:

`predict` *newvar*`, latent(M1[school])`

If you predict all the latent variables with the *stub*`*` trick, any multilevel latent variables will be automatically included among the new variables.

The remaining predictions are for models with categorical latent variables.

17. `predict` *newvar*`, classpr class(`*lcspec*`)` predicts the probability of belonging to the specified latent class. This probability is the same one used to combine the class specific densities to form the marginal likelihood.

18. `predict` *newvar*`, classposteriorpr class(`*lcspec*`)` predicts the posterior probability of belonging to the specified latent class. This probability is a function of the above latent class computed by `predict, classpr` and the fitted outcome densities.

19. You can use the same *stub*∗ trick mentioned above to obtain predicted probabilities for all latent classes. Replace *newvar* with *stub*∗ and omit `class()` to produce

```
predict stub*, classpr
```

or

```
predict stub*, classposteriorpr
```

See [SEM] **Example 28g**, [SEM] **Example 29g**, [SEM] **Example 50g**, [SEM] **Example 52g**, and [SEM] **predict after gsem**.

Using contrast, pwcompare, and margins (sem and gsem)

`contrast`, `pwcompare`, and `margins` are postestimation commands of Stata. All three can be used after `gsem`, and `margins` can be used after `sem` as well. Even so, these commands work best when you specify your model with Stata's factor-variable notation; see [SEM] **Intro 3**. `sem` does not allow that notation, and so obtaining desired results is a little more work.

1. Command `contrast` can be used after `gsem` to test linear hypotheses and make comparisons involving factor variables and their interactions. `contrast` allows for performing ANOVA-style tests of main effects, simple effects, or interaction effects. In addition, effects can be decomposed into specific types of contrasts such as comparison against the grand mean, orthogonal polynomial contrasts, and more. See [R] **contrast**.
2. Command `pwcompare` can be used after `gsem` to obtain all pairwise comparisons of marginal linear predictions, including cell means, marginal means, etc. See [R] **pwcompare**.
3. Command `margins` is used after `sem` or `gsem` to obtain margins, meaning estimated marginal means, predictive margins, adjusted predictions, and marginal effects. `margins` also allows for computing contrasts and pairwise comparisons of both linear and nonlinear predictions. See [SEM] **Example 50g**, [R] **margins**, [R] **margins, contrast**, and [R] **margins, pwcompare**.

Accessing stored results

`sem` and `gsem` store all results in `e()`; see *Stored results* in [SEM] **sem** and *Stored results* in [SEM] **gsem**. To get some idea of what is stored in `e()` after `sem` or `gsem` estimation, type

```
. ereturn list
  (output omitted)
```

You can save estimation results in files or temporarily in memory and do other useful things with them; see [R] **estimates**.

Not stored by `sem` in `e()` are the Bentler–Weeks matrices, but they can be obtained from the `r()` stored results of `estat framework`. (The Bentler–Weeks matrices are not relevant in the case of `gsem`.)

See [SEM] **sem** and [SEM] **estat framework**.

Reference

Satorra, A., and P. M. Bentler. 1994. "Corrections to test statistics and standard errors in covariance structure analysis". In *Latent Variables Analysis: Applications for Developmental Research*, edited by A. von Eye and C. C. Clogg, 399–419. Thousand Oaks, CA: Sage.

Also see

[SEM] **Intro 6** — Comparing groups

[SEM] **Intro 8** — Robust and clustered standard errors