### Syntax

```
gsem paths [if] [in] [ , options ]
```

where `paths` are the paths of the model in command-language path notation; see [SEM] `sem` and `gsem` path notation.

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>model_description_options</code></td>
<td>fully define, along with <code>paths</code>, the model to be fit</td>
</tr>
<tr>
<td><code>estimation_options</code></td>
<td>method used to obtain estimation results</td>
</tr>
<tr>
<td><code>reporting_options</code></td>
<td>reporting of estimation results</td>
</tr>
<tr>
<td><code>syntax_options</code></td>
<td>controlling interpretation of syntax</td>
</tr>
</tbody>
</table>

Factor variables and time-series operators are allowed. bootstrap, by, jackknife, permute, and statsby are allowed; see [U] 11.1.10 Prefix commands. Also see [SEM] `gsem postestimation` for features available after estimation.

### Menu

Statistics > SEM (structural equation modeling) > Model building and estimation

### Description

`gsem` fits generalized SEMs. When you use the Builder in `gsem` mode, you are using the `gsem` command.

### Options

`model_description_options` describe the model to be fit. The model to be fit is fully specified by `paths—which appear immediately after `gsem—and the options covariance( ), variance( ), and means( ). See [SEM] `gsem model description options` and [SEM] `sem` and `gsem` path notation.

`estimation_options` control how the estimation results are obtained. These options control how the standard errors (VCE) are obtained and control technical issues such as choice of estimation method. See [SEM] `gsem estimation options`.

`reporting_options` control how the results of estimation are displayed. See [SEM] `gsem reporting options`.

`syntax_options` control how the syntax that you type is interpreted. See [SEM] `sem` and `gsem` syntax options.
Remarks and examples

gsem provides important features not provided by sem and correspondingly omits useful features provided by sem. The differences in capabilities are the following:

1. gsem allows generalized linear response functions as well as the linear response functions allowed by sem.
2. gsem allows for multilevel models, something sem does not.
3. gsem allows Stata’s factor-variable notation to be used in specifying models, something sem does not.
4. gsem’s method ML is sometimes able to use more observations in the presence of missing values than can sem’s method ML. Meanwhile, gsem does not provide the MLMV method provided by sem for explicitly handling missing values.
5. gsem lacks the group-comparison capabilities that sem provides.
6. gsem lacks the ability to produce estimates adjusted for survey sampling that sem provides.
7. gsem cannot produce standardized coefficients.
8. gsem cannot use summary statistic datasets (SSDs); sem can.

gsem has nearly identical syntax to sem. Differences in syntax arise because of differences in capabilities. The resulting differences in syntax are the following:

1. gsem adds new syntax to paths to handle latent variables associated with multilevel modeling.
2. gsem adds new options to handle the family and link of generalized linear responses.
3. gsem deletes options related to features it does not have, such as group-comparison capabilities and SSDs.
4. gsem adds technical options for controlling features not provided by sem, such as numerical integration (quadrature choices), number of integration points, and a number of options dealing with starting values, which are a more difficult proposition in the generalized SEM framework.

For a readable explanation of what gsem can do and how to use it, see the intro sections. You might start with [SEM] intro 1.

For examples of gsem in action, see the example sections. You might start with [SEM] example 1.

For detailed syntax and descriptions, see the references below.

Remarks on three advanced topics are presented under the following headings:

Default normalization constraints
Default covariance assumptions
How to solve convergence problems

Default normalization constraints

gsem applies the same rules as sem to identify models; see [SEM] sem and see [SEM] intro 4. Everything said there about latent variables applies to multilevel latent variables such as M1[school] and M2[school>teacher].
gsem — Generalized structural equation model estimation command

Default covariance assumptions

gsem assumes the same covariance structures as does sem; see [SEM] sem and see [SEM] intro 4. gsem, however, treats covariances between observed exogenous variables as given. Actually, so does sem unless you specify an override. The override cannot be specified with gsem.

How to solve convergence problems

See [SEM] intro 12.

Stored results

gsem stores the following in e():

Scalars
- e(N) number of observations
- e(N_clust) number of clusters
- e(k) number of parameters
- e(k_cat#) number of categories for the #th depvar, ordinal
- e(k_dv) number of dependent variables
- e(k_eq) number of equations in e(b)
- e(k_out#) number of outcomes for the #th depvar, mlogit
- e(k_rc) number of covariances
- e(k_rs) number of variances
- e(ll) log likelihood
- e(n_quad) number of integration points
- e(rank) rank of e(V)
- e(ic) number of iterations
- e(rc) return code
- e(converged) 1 if target model converged, 0 otherwise

Macros
- e(cmd) gsem
- e(cmdline) command as typed
- e(depvar) names of dependent variables
- e(title) title in estimation output
- e(clustvar) name of cluster variable
- e(family#) family for the #th depvar
- e(link#) link for the #th depvar
- e(offset#) offset for the #th depvar
- e(intmethod) integration method
- e(vce) vcetype specified in vce()
- e(vcetype) title used to label Std. Err.
- e(opt) type of optimization
- e(which) max or min; whether optimizer is to perform maximization or minimization
- e(method) estimation method: ml
- e(ml_method) type of ml method
- e(user) name of likelihood-evaluator program
- e(technique) maximization technique
- e(datasignature) the checksum
- e(datasignaturevars) variables used in calculation of checksum
- e(properties) b V
- e(estat_cmd) program used to implement estat
- e(predict) program used to implement predict
- e(covariates) list of covariates
- e(footnote) program used to implement the footnote display
- e(marginsnotok) predictions not allowed by margins
Matrices

- `e(b)` parameter vector
- `e(b_pclass)` parameter class
- `e(cat#)` categories for the #th depvar, ordinal
- `e(out#)` outcomes for the #th depvar, mlogit
- `e(Cns)` constraints matrix
- `e(ilog)` iteration log (up to 20 iterations)
- `e(gradient)` gradient vector
- `e(V)` covariance matrix of the estimators
- `e(V_modelbased)` model-based variance

Functions

- `e(sample)` marks estimation sample

Also see

- [SEM] intro 1 — Introduction
- [SEM] sem and gsem path notation — Command syntax for path diagrams
- [SEM] gsem path notation extensions — Command syntax for path diagrams
- [SEM] gsem model description options — Model description options
- [SEM] gsem estimation options — Options affecting estimation
- [SEM] gsem reporting options — Options affecting reporting of results
- [SEM] sem and gsem syntax options — Options affecting interpretation of syntax
- [SEM] gsem postestimation — Postestimation tools for gsem
- [SEM] methods and formulas for gsem — Methods and formulas