

Description

SEM stands for structural equation model. Structural equation modeling is

1. A notation for specifying SEMs.
2. A way of thinking about SEMs.
3. Methods for estimating the parameters of SEMs.

Stata's `sem` and `gsem` commands fit these models: `sem` fits standard linear SEMs, and `gsem` fits generalized SEMs.

In `sem`, responses are continuous and models are linear regression.

In `gsem`, responses are continuous or binary, ordinal, count, or multinomial. Models are linear regression, gamma regression, logit, probit, ordinal logit, ordinal probit, Poisson, negative binomial, multinomial logit, and more.

`sem` fits models to single-level data.

`gsem` fits models to single-level or multilevel data. Latent variables can be included at any level. `gsem` can fit models with mixed effects, including random effects such as unobserved effects within patient, nested effects such as unobserved effects within patient within doctor, and crossed effects such as unobserved effects within occupation and country.

Meanwhile, `sem` provides features not provided by `gsem`: easy testing for whether groups such as males and females differ; estimation using observations with missing values under the assumption of joint normality; goodness-of-fit statistics, modification indices, tests of indirect effects, and more; and models fit using summary-statistic data.

There is obviously overlap between the capabilities of `sem` and `gsem`. In such cases, results will be nearly equal. Results should be exactly equal because both commands are producing estimates of the same mathematical model, but `sem` and `gsem` use different numerical machinery. `sem`'s machinery requires less calculation and fewer approximations and so is faster and slightly more accurate.

Remarks and examples

Structural equation modeling encompasses a broad array of models from linear regression to measurement models to simultaneous equations, including along the way confirmatory factor analysis (CFA), correlated uniqueness models, latent growth models, multiple indicators and multiple causes (MIMIC) models, and item response theory (IRT) models.

Structural equation modeling is not just an estimation method for a particular model in the way that Stata's `regress` and `probit` commands are, or even in the way that `stcox` and `mixed` are. Structural equation modeling is a way of thinking, a way of writing, and a way of estimating.

If you read the introductory manual pages in the front of this manual—[SEM] [intro 2](#), [SEM] [intro 3](#), and so on—we will do our best to familiarize you with SEM and our implementation of it.

Beginning with [SEM] [intro 2](#), entitled *Learning the language: Path diagrams and command language*, you will learn that

1. A particular SEM is usually described using a path diagram.
2. The `sem` and `gsem` commands allow you to use path diagrams to input models. In fact, the `sem` and `gsem` share the same GUI, called the SEM Builder.
3. `sem` and `gsem` alternatively allow you to use a command language to input models. The command language is similar to the path diagrams.

[SEM] [intro 3](#), entitled *Learning the language: Factor-variable notation (gsem only)*, amounts to a continuation of [SEM] [intro 2](#).

4. We teach you Stata's factor-variable notation, a wonderfully convenient shorthand for including categorical variables in models.

In [SEM] [intro 4](#), entitled *Substantive concepts*, you will learn that

5. `sem` provides four different estimation methods; you need to specify the method appropriate for the assumptions you are willing to make. For `gsem`, there are two estimation methods.
6. There are four types of variables in SEMs: A variable is observed or latent, and simultaneously it is endogenous or exogenous. To this, `sem` and `gsem` add another type of variable, the error variable. Error variables are latent exogenous variables with a fixed-unit path coefficient, and they are associated with a single endogenous variable. Error variables are denoted with an `e.` prefix, so if `y1` is an endogenous variable, then `e.y1` is the associated error variable.
7. It is easy to specify path constraints in SEMs—you just draw them, or omit drawing them, on the diagram. It is similarly easy with the SEM Builder as well as with `sem`'s and `gsem`'s command language.
8. Determining whether an SEM is identified can be difficult. We show you how to let the software check for you.
9. Identification also includes normalization constraints. `sem` and `gsem` apply normalization constraints automatically, but you can control that if you wish. Sometimes you might even need to control it.

In [SEM] [intro 5](#), entitled *Tour of models*,

10. We take you on a whirlwind tour of some of the models that `sem` and `gsem` can fit. This is a fun and useful section because we give you an overview without getting lost in the details.

Then in [SEM] [intro 6](#), entitled *Comparing groups (sem only)*,

11. We show you a highlight of `sem`: its ability to take an SEM and data consisting of groups—sexes, age categories, and the like—and fit the model in an interacted way that makes it easy for you to test whether and how the groups differ.

In [SEM] [intro 7](#), entitled *Postestimation tests and predictions*,

12. We show you how to redisplay results (`sem` and `gsem`), how to obtain exponentiated coefficients (`gsem` only), and how to obtain standardized results (`sem` only).
13. We show you how to obtain goodness-of-fit statistics (`sem` only).
14. We show you how to perform hypothesis tests, including tests for omitted paths, tests for relaxing constraints, and tests for model simplification.
15. We show you how to display other results, statistics, and tests.

16. We show you how to obtain predictions of observed response variables and predictions of latent variables. With `gsem`, you can obtain predicted means, probabilities, or counts that take into account the predictions of the latent variables, or you can set the latent variables to 0.

17. We show you how to access stored results.

In [SEM] [intro 8](#), entitled *Robust and clustered standard errors*,

18. We mention that `sem` and `gsem` optionally provide robust standard errors and provide clustered standard errors, which relaxes the assumption of independence of observations (or subjects) to independence within clusters of observations (subjects).

In [SEM] [intro 9](#), entitled *Standard errors, the full story*,

19. We provide lots of technical detail expanding on item 18.

In [SEM] [intro 10](#), entitled *Fitting models with survey data*,

20. We explain how `sem` and `gsem` can be used with Stata's `svy:` prefix to obtain results adjusted for complex survey designs, including clustered sampling and stratification.

In [SEM] [intro 11](#), entitled *Fitting models with summary statistics data (sem only)*,

21. We show you how to use `sem` with summary statistics data such as the correlation or covariance matrix rather than the raw data. Many sources, especially textbooks, publish data in summary statistics form.

Finally, in [SEM] [intro 12](#), entitled *Convergence problems and how to solve them*,

22. We regretfully inform you that some SEMs have difficulty converging. We figure 5% to 15% of complicated models will cause difficulty. We show you what to do and it is not difficult.

In the meantime,

23. There are many examples that we have collected for you in [SEM] [example 1](#), [SEM] [example 2](#), and so on. It is entertaining and informative simply to read the examples in order.

24. There is an alphabetical glossary in [SEM] [Glossary](#), located at the end of the manual.

If you prefer, you can skip all of this introductory material and go for the details. For the full experience, go directly to [SEM] [sem](#) and [SEM] [gsem](#). You will have no idea what we are talking about—we promise.

The technical sections, in logical order, are

Estimation

- [SEM] [sem](#)
- [SEM] [gsem](#)
- [SEM] [sem and gsem path notation](#)
- [SEM] [sem path notation extensions](#)
- [SEM] [gsem path notation extensions](#)
- [SEM] [Builder](#)
- [SEM] [Builder, generalized](#)
- [SEM] [sem model description options](#)
- [SEM] [gsem model description options](#)
- [SEM] [sem group options](#)
- [SEM] [sem ssd options](#)
- [SEM] [sem estimation options](#)
- [SEM] [gsem estimation options](#)
- [SEM] [sem reporting options](#)
- [SEM] [gsem reporting options](#)
- [SEM] [sem and gsem syntax options](#)
- [SEM] [sem option noxconditional](#)
- [SEM] [sem option select\(\)](#)
- [SEM] [sem and gsem option covstructure\(\)](#)
- [SEM] [sem option method\(\)](#)
- [SEM] [sem and gsem option reliability\(\)](#)
- [SEM] [sem and gsem option from\(\)](#)
- [SEM] [sem and gsem option constraints\(\)](#)
- [SEM] [gsem family-and-link options](#)
- [SEM] [ssd \(sem only\)](#)

Postestimation, summary of

- [SEM] [sem postestimation](#)
- [SEM] [gsem postestimation](#)

Reporting results

- [R] [estat](#)
- [SEM] [estat eform](#) (gsem only)
- [SEM] [estat teffects](#) (sem only)
- [SEM] [estat residuals](#) (sem only)
- [SEM] [estat framework](#) (sem only)

Goodness-of-fit tests

- [SEM] [estat gof](#) (sem only)
- [SEM] [estat eqgof](#) (sem only)
- [SEM] [estat ggof](#) (sem only)
- [R] [estat](#)

Hypotheses tests

[\[SEM\] estat mindices](#) (sem only)
[\[SEM\] estat eqtest](#) (sem only)
[\[SEM\] estat scoretests](#) (sem only)
[\[SEM\] estat ginvariant](#) (sem only)
[\[SEM\] estat stable](#) (sem only)
[\[SEM\] test](#)
[\[SEM\] lrtest](#)
[\[SEM\] testnl](#)
[\[SEM\] estat stdize](#) (sem only)

Linear and nonlinear combinations of results

[\[SEM\] lincom](#)
[\[SEM\] nlcom](#)

Predicted values

[\[SEM\] predict after sem](#)
[\[SEM\] predict after gsem](#)

Methods and formulas

[\[SEM\] methods and formulas for sem](#)
[\[SEM\] methods and formulas for gsem](#)

Many of these sections are technical, but mostly in the computer sense of the word. We suggest that when you read the technical sections, you skip to *Remarks and examples*. If you read the introductory sections, you will already know how to use the commands, so there is little reason to confuse yourself with syntax diagrams that are more precise than they are enlightening. However, the syntax diagrams do serve as useful reminders.

Also see

[\[SEM\] intro 2](#) — Learning the language: Path diagrams and command language
[\[SEM\] example 1](#) — Single-factor measurement model
[\[SEM\] Acknowledgments](#)