# sem and gsem option from() — Specifying starting values

Description Syntax Option Remarks and examples Also see

## Description

See [SEM] **Intro 12** for a description of starting values.

Starting values are usually not specified. When there are convergence problems, it is often necessary to specify starting values. You can specify starting values by using

1. suboption `init()` as described in [SEM] **sem and gsem path notation**, or by using
2. option `from()` as described here.

Option `from()` is especially convenient for using the solution of one model as starting values for another.

## Syntax

{`sem` | `gsem`} ... [, ... `from(`*matname*[, `skip`]`)` ... ]

{`sem` | `gsem`} ... [, ... `from(`*svlist*`)` ... ]

where *matname* is the name of a Stata matrix and

where *svlist* is a space-separated list of the form

*eqname*`:`*name* = #

## Option

`skip` is an option of `from(`*matname*`)`. It specifies to ignore any parameters in *matname* that do not appear in the model being fit. If this option is not specified, the existence of such parameters causes `sem` (`gsem`) to issue an error message.

## Remarks and examples

Remarks are presented under the following headings:

*Syntax 1, especially useful when dealing with convergence problems*
*Syntax 2, seldom used*

Option `from()` can be used with `sem` or `gsem`. We illustrate below using `sem`.

### Syntax 1, especially useful when dealing with convergence problems

Say you are attempting to fit

```
. sem your_full_model, ...
```

and are having difficulty with convergence. Following the advice in [SEM] **Intro 12**, you have simplified your model,

```
. sem your_simple_model, ...
```

and that does converge. You now want to use those values as starting values for the full model. Let's imagine that there are 47 estimated parameters in the simple model.

Using the standard `init()` method for specifying starting values, you now have a real job in front of you. You have to type your full model, find all the places where you want to add starting values, and add an `init()` suboption. Just a piece of your full model might read

```
... (y<-L1 L2) (L1->x1 x2) (L2->x3 L4) ...
```

and you need to modify that to read

```
... (y<-(L1, init(14.283984)) L2)           ///
  (L1->(x1, init(2.666532)) (x2, init(-6.39499))) ///
  (L2->x3 L4) ...
```

That change handles 3 of the 47 parameters you need to specify.

There is an easier way. Type

```
. sem your_simple_model, ...
. matrix b = e(b)
. sem your_full_model, ... from(b)
```

Here is how this works:

1. You fit the simple model. `sem` stores the resulting parameters in `e(b)`.
2. You store the fitted parameters in Stata matrix `b`.
3. To fit your full model, type the model just as you would usually, and add option `from(b)`. That option tells `sem` to get any starting values it can from Stata matrix `b`. `sem` does that and then follows its usual logic for producing starting values for the remaining parameters.

Just because you use `from(b)` does not mean you cannot specify starting values with `init()`. You can even specify starting values for some of the same parameters. Starting values specified by `init()` take precedence over those obtained from `from()`.

## Syntax 2, seldom used

In syntax 2, you specify

```
. sem ..., ... from(eqname:name = # eqname:name = # ...)
```

For instance, you could type

```
. sem ..., ... from(var(X):_cons=10)
```

or you could type

```
. sem ..., ... var(X, init(10))
```

It is usually easier to type the second. See [SEM] **sem and gsem path notation**.

You may combine the two notations. If starting values are specified for a parameter both ways, those specified by `init()` take precedence.

## Also see

[SEM] **sem** — Structural equation model estimation command

[SEM] **gsem** — Generalized structural equation model estimation command

[SEM] **sem and gsem path notation** — Command syntax for path diagrams

[SEM] **gsem model description options** — Model description options

[SEM] **sem model description options** — Model description options

[SEM] **gsem estimation options** — Options affecting estimation

[R] **Maximize** — Details of iterative maximization

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.