Title

stata.com

sem — Structural equation model estimation command

Description

`sem` fits structural equation models. Even when you use the SEM Builder, you are using the `sem` command.

Menu

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

Syntax

`sem` *paths* [*if*] [*in*] [*weight*] [, *options*]

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

options	Description
model_description_options	fully define, along with *paths*, the model to be fit
group_options	fit model for different groups
ssd_options	for use with summary statistics data
estimation_options	method used to obtain estimation results
reporting_options	reporting of estimation results
syntax_options	controlling interpretation of syntax

Time-series operators are allowed.

`bootstrap`, `by`, `collect`, `jackknife`, `permute`, `statsby`, and `svy` are allowed; see **[U] 11.1.10 Prefix commands**.

Weights are not allowed with the `bootstrap` prefix; see [R] **bootstrap**.

`vce()` and weights are not allowed with the `svy` prefix; see [SVY] **svy**.

`fweights`, `iweights`, and `pweights` are allowed; see **[U] 11.1.6 weight**.

Also see [SEM] **sem postestimation** for features available after estimation.

Options

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

group_options allow the specified model to be fit for different subgroups of the data, with some parameters free to vary across groups and other parameters constrained to be equal across groups. See [SEM] **sem group options**.

ssd_options allow models to be fit using summary statistics data (SSD), meaning data on means, variances (standard deviations), and covariances (correlations). See [SEM] **sem ssd options**.

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] **sem estimation options**.

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

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

Remarks and examples

stata.com

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

For examples of `sem` in action, see any of 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

`sem` applies the following rules as necessary to identify the model:

1. `means(1: LatentExogenous@0)`
   `sem` constrains all latent exogenous variables to have mean 0. When the `group()` option is specified, this rule is applied to the first group only.
2. `(LatentEndogenous <- _cons@0)`
   `sem` sets all latent endogenous variables to have intercept 0.
3. `(first <- Latent@1)`
   If latent variable `Latent` is measured by observed endogenous variables, then `sem` sets the path coefficient of `(first<-Latent)` to be 1; `first` is the first observed endogenous variable.
4. `(First<-Latent@1)`
   If item 3 does not apply—if latent variable `Latent` is measured by other latent endogenous variables only—then `sem` sets the path coefficient of `First<-Latent` to be 1; `First` is the first latent variable.

The above constraints are applied as needed. Here are the available overrides:

1. To override the normalization constraints, specify your own constraints. Most normalization constraints are added by `sem` as needed. See *How sem (gsem) solves the problem for you* under *Identification 2: Normalization constraints (anchoring)* in [SEM] **Intro 4**.

2. To override means() constraints, you must use the means() option to free the parameter. To override that the mean of latent exogenous variable MyLatent has mean 0, specify the means(MyLatent) option. See [SEM] **sem and gsem path notation**.
3. To override constrained path coefficients from _cons, such as (LatentEndogenous <- _cons@0), you must explicitly specify the path without a constraint (LatentEndogenous <- _cons). See [SEM] **sem and gsem path notation**.

Default covariance assumptions

sem assumes the following covariance structure:

1. covstructure(_Ex, unstructured)
   All exogenous variables (observed and latent) are assumed to be correlated with each other.
2. covstructure(e._En, diagonal)
   The error variables associated with all endogenous variables are assumed to be uncorrelated with each other.

You can override these assumptions by

1. Constraining or specifying the relevant covariance to allow e.y and e.Ly to be correlated (specify the covariance(e.y*e.Ly) option); see [SEM] **sem model description options**.
2. Using the covstructure() option; see [SEM] **sem and gsem option covstructure( )**.

How to solve convergence problems

See [SEM] **Intro 12**.

Stored results

sem stores the following in e():

Scalars

e(N)	number of observations
e(N_clust)	number of clusters
e(N_groups)	number of groups
e(N_missing)	number of missing values in the sample for method(mlmv)
e(ll)	log likelihood of model
e(df_m)	model degrees of freedom
e(df_b)	baseline model degrees of freedom
e(df_s)	saturated model degrees of freedom
e(chi2_ms)	test of target model against saturated model
e(df_ms)	degrees of freedom for e(chi2_ms)
e(p_ms)	p-value for e(chi2_ms)
e(chi2sb_ms)	Satorra–Bentler scaled test of target model against saturated model
e(psb_ms)	p-value for e(chi2sb_ms)
e(sbc_ms)	Satorra–Bentler correction factor for e(chi2sb_ms)
e(chi2_bs)	test of baseline model against saturated model
e(df_bs)	degrees of freedom for e(chi2_bs)
e(p_bs)	p-value for e(chi2_bs)
e(chi2sb_bs)	Satorra–Bentler scaled test of baseline model against saturated model
e(psb_bs)	p-value for e(chi2sb_bs)
e(sbc_bs)	Satorra–Bentler correction factor for e(chi2_bs)
e(rank)	rank of e(V)
e(ic)	number of iterations
e(rc)	return code

e(converged)	1 if target model converged, 0 otherwise
e(critvalue)	log likelihood or discrepancy of fitted model
e(critvalue_b)	log likelihood or discrepancy of baseline model
e(critvalue_s)	log likelihood or discrepancy of saturated model
e(modelmeans)	1 if fitting means and intercepts, 0 otherwise

Macros

e(cmd)	sem
e(cmdline)	command as typed
e(data)	raw or ssd if SSD were used
e(wtype)	weight type
e(wexp)	weight expression
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(method)	estimation method: ml, mlmv, or adf
e(technique)	maximization technique
e(properties)	b V
e(estat_cmd)	program used to implement estat
e(predict)	program used to implement predict
e(lyvars)	names of latent y variables
e(oyvars)	names of observed y variables
e(lxvars)	names of latent x variables
e(oxvars)	names of observed x variables
e(groupvar)	name of group variable
e(xconditional)	empty if noxconditional specified, xconditional otherwise
e(marginsnotok)	predictions not allowed by margins
e(marginsdefault)	default predict() specification for margins

Matrices

e(b)	parameter vector
e(b_std)	standardized parameter vector
e(b_pclass)	parameter class
e(Cns)	constraints matrix
e(admissible)	admissibility of Σ, Ψ, Φ
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	covariance matrix of the estimators
e(V_std)	standardized covariance matrix of the estimators
e(V_modelbased)	model-based variance
e(nobs)	vector with number of observations per group
e(groupvalue)	vector of group values of e(groupvar)

Functions

e(sample)	marks estimation sample (not with SSD)

In addition to the above, the following is stored in r():

Matrices

r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

References

Baldwin, S. 2019. *Psychological Statistics and Psychometrics Using Stata*. College Station, TX: Stata Press.

MacDonald, K. 2016. Group comparisons in structural equation models: Testing measurement invariance. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2016/08/23/group-comparisons-in-structural-equation-models-testing-measurement-invariance/.

Wiggins, V. L. 2011. Multilevel random effects in xtmixed and sem—the long and wide of it. *The Stata Blog: Not Elsewhere Classified*. http://blog.stata.com/2011/09/28/multilevel-random-effects-in-xtmixed-and-sem-the-long-and-wide-of-it/.

Also see

[SEM] **Intro 1** — Introduction

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

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

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

[SEM] **sem group options** — Fitting models on different groups

[SEM] **sem ssd options** — Options for use with summary statistics data

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

[SEM] **sem reporting options** — Options affecting reporting of results

[SEM] **sem and gsem syntax options** — Options affecting interpretation of syntax

[SEM] **sem postestimation** — Postestimation tools for sem

[SEM] **Methods and formulas for sem** — Methods and formulas for sem

[SVY] **svy estimation** — Estimation commands for survey data

[U] **20 Estimation and postestimation commands**