Title

example 20 — Two-factor measurement model by group

Description   Remarks and examples   Reference   Also see

Description

Below we demonstrate `sem`'s `group()` option, which allows fitting models in which path coefficients and covariances differ across groups of the data, such as for males and females. We use the following data:

```
. use http://www.stata-press.com/data/r14/sem_2fmmby
(two-factor CFA)

. ssd describe

  Summary statistics data from
  http://www.stata-press.com/data/r14/sem_2fmmby.dta
    obs:             385                  two-factor CFA
   vars:              16                  25 May 2014 11:11
                                          (_dta has notes)
  ------------------------------------------------------------------------
  variable name                  variable label
  ------------------------------------------------------------------------
  phyab1                         Physical ability 1
  phyab2                         Physical ability 2
  phyab3                         Physical ability 3
  phyab4                         Physical ability 4
  appear1                        Appearance 1
  appear2                        Appearance 2
  appear3                        Appearance 3
  appear4                        Appearance 4
  peerrel1                       Relationship w/ peers 1
  peerrel2                       Relationship w/ peers 2
  peerrel3                       Relationship w/ peers 3
  peerrel4                       Relationship w/ peers 4
  parrel1                        Relationship w/ parent 1
  parrel2                        Relationship w/ parent 2
  parrel3                        Relationship w/ parent 3
  parrel4                        Relationship w/ parent 4
  ------------------------------------------------------------------------
  Group variable:  grade  (2 groups)
   Obs. by group:  134, 251

. notes

_dta:
  1.  Summary statistics data from Marsh, H. W. and Hocevar, D., 1985,
      "Application of confirmatory factor analysis to the study of
      self-concept: First- and higher order factor models and their invariance
      across groups", _Psychological Bulletin_, 97: 562-582.
  2.  Summary statistics based on 134 students in grade 4 and 251 students in
      grade 5 from Sydney, Australia.
  3.  Group 1 is grade 4, group 2 is grade 5.
  4.  Data collected using the Self-Description Questionnaire and includes
      sixteen subscales designed to measure nonacademic status: four intended
      to measure physical ability, four intended to measure physical
      appearance, four intended to measure relations with peers, and four
      intended to measure relations with parents.
```

Remarks and examples

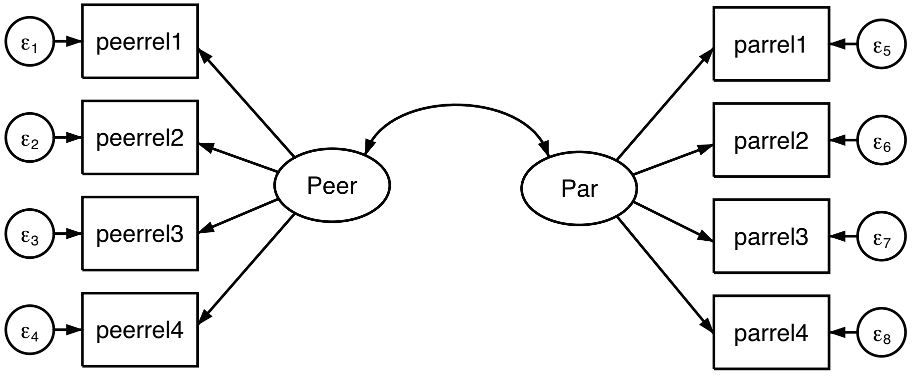
stata.com

Remarks are presented under the following headings:

Background

See [SEM] **intro 6** for background on `sem`'s `group()` option.

We will fit the model



which in command syntax can be written as

```
(Peer -> peerrel1 peerrel2 peerrel3 peerrel4)   ///
(Par  -> parrel1  parrel2  parrel3  parrel4)
```

We are using the same data used in [SEM] **example 15**, but we are using more of the data and fitting a different model. To remind you, those data were collected from students in grade 5. The dataset we are using, however, has data for students from grade 4 and from grade 5, which was created in [SEM] **example 19**. We have the following observed variables:

1. Four measures of physical ability.
2. Four measures of appearance.
3. Four measures of quality of relationship with peers.
4. Four measures of quality of relationship with parents.

In this example, we will consider solely the measurement problem, and include only the measurement variables for the two kinds of relationship quality. We are going to treat quality of relationship with peers as measures of underlying factor `Peer` and quality of relationship with parents as measures of underlying factor `Par`.

Below we will

1. Fit the model with all the data. This amounts to assuming that the students in grades 4 and 5 are identical in terms of this measurement problem.
2. Fit the model with `sem`'s `group()` option, which will constrain some parameters to be the same for students in grades 4 and 5 and leave free of constraint the others.

Fitting the model with all the data

Throughout this example, we want you to appreciate that we are using SSD and that matters not at all. Not one command would have a different syntax or option, or produce a different result, if we had the real data.

We begin by fitting the model with all the data:

```
. sem (Peer -> peerrel1 peerrel2 peerrel3 peerrel4)
>     (Par  -> parrel1  parrel2  parrel3  parrel4)
Endogenous variables

Measurement:  peerrel1 peerrel2 peerrel3 peerrel4 parrel1 parrel2 parrel3 parrel4

Exogenous variables

Latent:       Peer Par

Fitting target model:

Iteration 0:   log likelihood =  -5559.545
Iteration 1:   log likelihood =  -5558.609
Iteration 2:   log likelihood = -5558.6017
Iteration 3:   log likelihood = -5558.6017

Structural equation model                       Number of obs     =        385
Estimation method  = ml
Log likelihood     = -5558.6017

 ( 1)  [peerrel1]Peer = 1
 ( 2)  [parrel1]Par = 1
```

	Coef.	OIM Std. Err.	z	P>|z|	[95% Conf.	Interval]
Measurement						
peerr~1 <-						
Peer	1	(constrained)				
_cons	8.681221	.0937197	92.63	0.000	8.497534	8.864908
peerr~2 <-						
Peer	1.113865	.09796	11.37	0.000	.9218666	1.305863
_cons	7.828623	.1037547	75.45	0.000	7.625268	8.031979
peerr~3 <-						
Peer	1.42191	.114341	12.44	0.000	1.197806	1.646014
_cons	7.359896	.1149905	64.00	0.000	7.134519	7.585273
peerr~4 <-						
Peer	1.204146	.0983865	12.24	0.000	1.011312	1.39698
_cons	8.150779	.1023467	79.64	0.000	7.950183	8.351375
parrel1 <-						
Par	1	(constrained)				
_cons	9.339558	.0648742	143.96	0.000	9.212407	9.46671
parrel2 <-						
Par	1.112383	.1378687	8.07	0.000	.8421655	1.382601
_cons	9.220494	.0742356	124.21	0.000	9.074994	9.365993
parrel3 <-						
Par	2.037924	.204617	9.96	0.000	1.636882	2.438966
_cons	8.676961	.088927	97.57	0.000	8.502667	8.851255
parrel4 <-						
Par	1.52253	.1536868	9.91	0.000	1.221309	1.82375
_cons	9.045247	.0722358	125.22	0.000	8.903667	9.186826

```
------------------------------------------------------------------------------------
var(e.peer~1) |  1.809309   .1596546                       1.521956    2.150916
var(e.peer~2) |  2.193804    .194494                       1.843884    2.610129
var(e.peer~3) |  1.911874    .214104                       1.535099    2.381126
var(e.peer~4) |  1.753037   .1749613                       1.441575    2.131792
var(e.parr~1) |  1.120333   .0899209                       .9572541    1.311193
var(e.parr~2) |  1.503003   .1200739                       1.285162    1.757769
var(e.parr~3) |  .9680081   .1419777                       .7261617    1.290401
var(e.parr~4) |  .8498834   .0933687                        .685245    1.054078
    var(Peer) |  1.572294   .2255704                       1.186904    2.082822
     var(Par) |  .5000022    .093189                       .3469983    .7204709
--------------+---------------------------------------------------------------------
cov(Peer,Par) |  .4226706   .0725253     5.83   0.000      .2805236    .5648176
------------------------------------------------------------------------------------
LR test of model vs. saturated: chi2(19)   =     28.19, Prob > chi2 = 0.0798
```

Note:

1. We are using SSD with data for two separate groups. There is no hint of that in the output above because `sem` combined the summary statistics and produced overall results just as if we had the real data.

Fitting the model with the group() option

```
. sem (Peer -> peerrel1 peerrel2 peerrel3 peerrel4)
>     (Par  -> parrel1  parrel2  parrel3  parrel4), group(grade)
Endogenous variables
Measurement:  peerrel1 peerrel2 peerrel3 peerrel4 parrel1 parrel2 parrel3
              parrel4
Exogenous variables
Latent:       Peer Par
Fitting target model:
Iteration 0:   log likelihood =  -13049.77  (not concave)
Iteration 1:   log likelihood = -10819.682  (not concave)
Iteration 2:   log likelihood = -8873.4568  (not concave)
Iteration 3:   log likelihood = -6119.7114  (not concave)
Iteration 4:   log likelihood =  -5949.354  (not concave)
Iteration 5:   log likelihood = -5775.6085  (not concave)
Iteration 6:   log likelihood = -5713.9178  (not concave)
Iteration 7:   log likelihood = -5638.1208  (not concave)
Iteration 8:   log likelihood = -5616.6335  (not concave)
Iteration 9:   log likelihood = -5595.7507  (not concave)
Iteration 10:  log likelihood = -5589.9802  (not concave)
Iteration 11:  log likelihood = -5578.8701  (not concave)
Iteration 12:  log likelihood = -5574.0162  (not concave)
Iteration 13:  log likelihood = -5568.0786
Iteration 14:  log likelihood = -5551.7349
Iteration 15:  log likelihood = -5544.0052
Iteration 16:  log likelihood = -5542.7113
Iteration 17:  log likelihood = -5542.6775
Iteration 18:  log likelihood = -5542.6774
Structural equation model                       Number of obs      =        385
Grouping variable  = grade                      Number of groups   =          2
Estimation method  = ml
Log likelihood     = -5542.6774
```

```
( 1)  [peerrel1]1bn.grade#c.Peer = 1
( 2)  [peerrel2]1bn.grade#c.Peer - [peerrel2]2.grade#c.Peer = 0
( 3)  [peerrel3]1bn.grade#c.Peer - [peerrel3]2.grade#c.Peer = 0
( 4)  [peerrel4]1bn.grade#c.Peer - [peerrel4]2.grade#c.Peer = 0
( 5)  [parrel1]1bn.grade#c.Par = 1
( 6)  [parrel2]1bn.grade#c.Par - [parrel2]2.grade#c.Par = 0
( 7)  [parrel3]1bn.grade#c.Par - [parrel3]2.grade#c.Par = 0
( 8)  [parrel4]1bn.grade#c.Par - [parrel4]2.grade#c.Par = 0
( 9)  [peerrel1]1bn.grade - [peerrel1]2.grade = 0
(10)  [peerrel2]1bn.grade - [peerrel2]2.grade = 0
(11)  [peerrel3]1bn.grade - [peerrel3]2.grade = 0
(12)  [peerrel4]1bn.grade - [peerrel4]2.grade = 0
(13)  [parrel1]1bn.grade - [parrel1]2.grade = 0
(14)  [parrel2]1bn.grade - [parrel2]2.grade = 0
(15)  [parrel3]1bn.grade - [parrel3]2.grade = 0
(16)  [parrel4]1bn.grade - [parrel4]2.grade = 0
(17)  [peerrel1]2.grade#c.Peer = 1
(18)  [parrel1]2.grade#c.Par = 1
(19)  [mean(Peer)]1bn.grade = 0
(20)  [mean(Par)]1bn.grade = 0
```

	Coef.	OIM Std. Err.	z	P>|z|	[95% Conf.	Interval]
Measurement						
peerr~1 <-						
Peer						
[*]	1	(constrained)				
_cons						
[*]	8.466539	.1473448	57.46	0.000	8.177748	8.755329
peerr~2 <-						
Peer						
[*]	1.109234	.0975279	11.37	0.000	.9180833	1.300385
_cons						
[*]	7.589872	.1632145	46.50	0.000	7.269977	7.909766
peerr~3 <-						
Peer						
[*]	1.409361	.1138314	12.38	0.000	1.186256	1.632467
_cons						
[*]	7.056996	.1964299	35.93	0.000	6.672001	7.441992
peerr~4 <-						
Peer						
[*]	1.195982	.0980272	12.20	0.000	1.003852	1.388112
_cons						
[*]	7.89358	.169158	46.66	0.000	7.562036	8.225123
parrel1 <-						
Par						
[*]	1	(constrained)				
_cons						
[*]	9.368654	.0819489	114.32	0.000	9.208037	9.529271
parrel2 <-						
Par						
[*]	1.104355	.1369365	8.06	0.000	.8359649	1.372746
_cons						
[*]	9.287629	.0903296	102.82	0.000	9.110587	9.464672

parrel3 <-						
Par						
[*]	2.05859	.2060583	9.99	0.000	1.654723	2.462457
_cons						
[*]	8.741898	.136612	63.99	0.000	8.474144	9.009653
parrel4 <-						
Par						
[*]	1.526706	.1552486	9.83	0.000	1.222424	1.830987
_cons						
[*]	9.096609	.1061607	85.69	0.000	8.888538	9.30468
mean(Peer)						
1	0	(constrained)				
2	.3296841	.1570203	2.10	0.036	.02193	.6374382
mean(Par)						
1	0	(constrained)				
2	-.0512439	.0818255	-0.63	0.531	-.211619	.1091313
var(e.peer~1)						
1	1.824193	.2739446			1.359074	2.448489
2	1.773813	.1889104			1.439644	2.185549
var(e.peer~2)						
1	2.236974	.3310875			1.673699	2.989817
2	2.165228	.2321565			1.75484	2.671589
var(e.peer~3)						
1	1.907009	.3383293			1.346908	2.700023
2	1.950679	.2586196			1.504298	2.529516
var(e.peer~4)						
1	1.639881	.272764			1.18367	2.271925
2	1.822448	.2151827			1.445942	2.296992
var(e.parr~1)						
1	.9669121	.1302489			.7425488	1.259067
2	1.213159	.1192634			1.000547	1.470949
var(e.parr~2)						
1	.9683878	.133192			.7395628	1.268012
2	1.79031	.1747374			1.478596	2.167739
var(e.parr~3)						
1	.8377567	.1986089			.526407	1.333258
2	1.015707	.1713759			.7297073	1.4138
var(e.parr~4)						
1	.8343032	.1384649			.6026352	1.15503
2	.8599648	.1165865			.6592987	1.121706
var(Peer)						
1	2.039297	.3784544			1.41747	2.933912
2	1.307976	.2061581			.9603661	1.781406
var(Par)						
1	.4492996	.1011565			.2889976	.6985183
2	.5201696	.1029353			.3529413	.7666329
cov(Peer,Par)						
1	.5012091	.1193333	4.20	0.000	.2673201	.7350982
2	.3867156	.079455	4.87	0.000	.2309867	.5424445

Note: [*] identifies parameter estimates constrained to be equal across groups.
LR test of model vs. saturated: chi2(50) = 61.91, Prob > chi2 = 0.1204

Notes:

1. In *Which parameters vary by default, and which do not* in [SEM] **intro 6**, we wrote that, generally speaking, when we specify `group(`*groupvar*`)`, the measurement part of the model is constrained by default to be the same across the groups, whereas the remaining parts will have separate parameters for each group.

   More precisely, we revealed that `sem` classifies each parameter into one of nine classes, which are the following:

Class description	Class name	
1. structural coefficients	`scoef`	
2. structural intercepts	`scons`	
3. measurement coefficients	`mcoef`	
4. measurement intercepts	`mcons`	
5. covariances of structural errors	`serrvar`	
6. covariances of measurement errors	`merrvar`	
7. covariances between structural and measurement errors	`smerrcov`	
8. means of exogenous variables	`meanex`	(*)
9. covariances of exogenous variables	`covex`	(*)
10. all the above	`all`	(*)
11. none of the above	`none`	

   (*) Be aware that classes 8, 9, and 10 (`meanex`, `covex`, and `all`) exclude the observed exogenous variables—include only the latent exogenous variables—unless you specify option `noxconditional` or the `noxconditional` option is otherwise implied; see [SEM] **sem option noxconditional**. This is what you would desire in most cases.

   By default, classes 3 and 4 are constrained to be equal and the rest are allowed to vary.

2. Thus you might expect that most of the parameters of our model would have been left unconstrained until you remember that we are fitting a measurement model. That is why `sem` listed 20 constraints at the top of the estimation results. Some of the constraints are substantive and some are normalization.

3. In the output, paths listed with an asterisk are constrained to be equal across groups.

   Paths labeled with group 1 and group 2 are group specific (unconstrained).

   In our data, group 1 corresponds with students in grade 4, and group 2 corresponds with students in grade 5.

4. It may surprise you that the output contains estimates for the means of the latent variables. Usually, `sem` does not report this.

   Usually, you are running on only one group of data and those means cannot be estimated, at least not without additional identifying constraints. When you are running on two or more groups, the means for all the groups except one can be estimated.

In [SEM] **example 21**, we use `estat ggof` to evaluate goodness of fit group by group.

In [SEM] **example 22**, we use `estat ginvariant` to test whether parameters that are constrained across groups should not be and whether parameters that are not constrained could be.

In [SEM] **example 23**, we show how to constrain the parameters we choose to be equal across groups.

Fitting the model with the Builder

Use the diagram above for reference.

1. Open the dataset.

   In the Command window, type

   ```
   . use http://www.stata-press.com/data/r14/sem_2fmmby
   ```

2. Open a new Builder diagram.

   Select menu item **Statistics > SEM (structural equation modeling) > Model building and estimation**.

3. Create the measurement component for relationships with peers.

   Select the Add Measurement Component tool, , and then click in the diagram about halfway down from the top and about one-third of the way in from the left.

   In the resulting dialog box,

   a. change the *Latent variable name* to `Peer`;

   b. select `peerrel1`, `peerrel2`, `peerrel3`, and `peerrel4` by using the *Measurement variables* control;

   c. select `Left` in the *Measurement direction* control;

   d. click on **OK**.

   If you wish, move the component by clicking on any variable and dragging it.

4. Create the measurement component for relationships with parents.

   Select the Add Measurement Component tool, , and then click in the diagram about halfway down from the top and about one-third of the way in from the right.

   In the resulting dialog box,

   a. change the *Latent variable name* to `Par`;

   b. select `parrel1`, `parrel2`, `parrel3`, and `parrel4` by using the *Measurement variables* control;

   c. select `Right` in the *Measurement direction* control;

   d. click on **OK**.

   If you wish, move the component by clicking on any variable and dragging it.

5. Correlate the latent variables.

   a. Select the Add Covariance tool, .

   b. Click in the upper-right quadrant of the `Peer` oval (it will highlight when you hover over it), and drag a covariance to the upper-left quadrant of the `Par` oval (it will highlight when you can release to connect the covariance).

6. Clean up.

   If you do not like where a covariance has been connected to its variable, use the Select tool, , to click on the covariance, and then simply click on where it connects to an oval and drag the endpoint. You can also change the bow of the covariance by dragging the control point that extends from one end of the selected covariance.

7. Estimate.

   Click on the **Estimate** button, , in the Standard Toolbar.

   In the resulting dialog box, do the following:

   a. Select the *Group* tab.

   b. Select the *Group analysis* radio button. The variable `grade` should appear in the *Group variable* control.

   c. Click on **OK**.

   d. In the Standard Toolbar, use the *Group* control to toggle between results for group 1 and group 2.

You can open a completed diagram in the Builder by typing

```
. webgetsem sem_2fmmby
```

Reference

Acock, A. C. 2013. *Discovering Structural Equation Modeling Using Stata*. Rev. ed. College Station, TX: Stata Press.

Also see

[SEM] **example 3** — Two-factor measurement model

[SEM] **example 19** — Creating multiple-group summary statistics data

[SEM] **example 21** — Group-level goodness of fit

[SEM] **example 22** — Testing parameter equality across groups

[SEM] **example 23** — Specifying parameter constraints across groups

[SEM] **intro 6** — Comparing groups (sem only)

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

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