

**example 8** — Testing that coefficients are equal, and constraining them

[Description](#)      [Remarks and examples](#)      [Also see](#)

## Description

This example continues where [\[SEM\] example 7](#) left off, where we typed

```
. use http://www.stata-press.com/data/r15/sem_sm1
. ssd describe
. notes
. sem (r_occasp <- f_occasp r_intel r_ses f_ses) ///
      (f_occasp <- r_occasp f_intel f_ses r_ses), ///
      cov(e.r_occasp*e.f_occasp) standardized
. estat stable
. estat teffects
```

## Remarks and examples

stata.com

Remarks are presented under the following headings:

[Using test to evaluate adding constraints](#)  
[Refitting the model with added constraints](#)  
[Using estat scoretests to test whether constraints can be relaxed](#)

We want to show you how to evaluate potential constraints after estimation, how to fit a model with constraints, and how to evaluate enforced constraints after estimation.

Obviously, in a real analysis, if you evaluated potential constraints after estimation, there would be no reason to evaluate enforced constraints after estimation, and vice versa.

## Using test to evaluate adding constraints

In this model of respondents and corresponding friends, it would be surprising if the coefficients relating friends' characteristics to respondents' occupational aspirations and vice versa were not equal. It would also be surprising if coefficients relating a respondent's characteristics to his occupational aspirations were not equal to those of his friends' characteristics to his occupational aspirations. The paths that we suspect should be equal are

```
r_intel  -> r_occasp      f_intel  -> f_occasp
r_ses    -> r_occasp      f_ses    -> f_occasp
f_ses    -> r_occasp      r_ses    -> f_occasp
f_occasp -> r_occasp      r_occasp -> f_occasp
```

You are about to learn that to test whether those paths have equal coefficients, you type

```
. test (_b[r_occasp:r_intel ]==_b[f_occasp:f_intel ]) ///
      (_b[r_occasp:r_ses  ]==_b[f_occasp:f_ses  ]) ///
      (_b[r_occasp:f_ses  ]==_b[f_occasp:r_ses  ]) ///
      (_b[r_occasp:f_occasp]==_b[f_occasp:r_occasp])
```

## 2 example 8 — Testing that coefficients are equal, and constraining them

In Stata, `_b[]` is how one accesses the estimated parameters. It is difficult to remember what the names are. To determine the names of the parameters, replay the `sem` results with the `coeflegend` option:

```
. sem, coeflegend
Structural equation model          Number of obs    =          329
Estimation method = ml
Log likelihood      = -2617.0489
```

---

|                              | Coef.     | Legend                          |
|------------------------------|-----------|---------------------------------|
| Structural                   |           |                                 |
| r_occasp                     |           |                                 |
| f_occasp                     | .2773441  | _b[r_occasp:f_occasp]           |
| r_intel                      | .2854766  | _b[r_occasp:r_intel]            |
| r_ses                        | .1570082  | _b[r_occasp:r_ses]              |
| f_ses                        | .0973327  | _b[r_occasp:f_ses]              |
| f_occasp                     |           |                                 |
| r_occasp                     | .2118102  | _b[f_occasp:r_occasp]           |
| r_ses                        | .0794194  | _b[f_occasp:r_ses]              |
| f_ses                        | .1681772  | _b[f_occasp:f_ses]              |
| f_intel                      | .3693682  | _b[f_occasp:f_intel]            |
| var(e.r_oc~p)                | .6868304  | _b[/var(e.r_occasp)]            |
| var(e.f_oc~p)                | .6359151  | _b[/var(e.f_occasp)]            |
| cov(e.r_oc~p,<br>e.f_occasp) | -.1536992 | _b[/cov(e.r_occasp,e.f_occasp)] |

---

```
LR test of model vs. saturated: chi2(0)    =          0.00, Prob > chi2 =          .
```

With the parameter names at hand, to perform the test, we can type

```
. test (_b[r_occasp:r_intel ]==_b[f_occasp:f_intel ])
>      (_b[r_occasp:r_ses  ]==_b[f_occasp:f_ses  ])
>      (_b[r_occasp:f_ses  ]==_b[f_occasp:r_ses  ])
>      (_b[r_occasp:f_occasp]==_b[f_occasp:r_occasp])
( 1) [r_occasp]r_intel - [f_occasp]f_intel = 0
( 2) [r_occasp]r_ses - [f_occasp]f_ses = 0
( 3) [r_occasp]f_ses - [f_occasp]r_ses = 0
( 4) [r_occasp]f_occasp - [f_occasp]r_occasp = 0
      chi2( 4) =      1.61
      Prob > chi2 =      0.8062
```

We cannot reject the constraint, just as we expected.

## Refitting the model with added constraints

We could refit the model with these constraints by typing

```
. sem (r_occasp <- f_occasp@b1 r_intel@b2 r_ses@b3 f_ses@b4)
>     (f_occasp <- r_occasp@b1 f_intel@b2 f_ses@b3 r_ses@b4),
>     cov(e.r_occasp*e.f_occasp)
```

Endogenous variables

Observed: r\_occasp f\_occasp

Exogenous variables

Observed: r\_intel r\_ses f\_ses f\_intel

Fitting target model:

Iteration 0: log likelihood = -2617.8735

Iteration 1: log likelihood = -2617.8705

Iteration 2: log likelihood = -2617.8705

Structural equation model Number of obs = 329

Estimation method = ml

Log likelihood = -2617.8705

( 1) [r\_occasp]f\_occasp - [f\_occasp]r\_occasp = 0

( 2) [r\_occasp]r\_intel - [f\_occasp]f\_intel = 0

( 3) [r\_occasp]r\_ses - [f\_occasp]f\_ses = 0

( 4) [r\_occasp]f\_ses - [f\_occasp]r\_ses = 0

|                              | OIM       |           |       |       | [95% Conf. Interval] |          |
|------------------------------|-----------|-----------|-------|-------|----------------------|----------|
|                              | Coef.     | Std. Err. | z     | P> z  |                      |          |
| <b>Structural</b>            |           |           |       |       |                      |          |
| r_occasp                     |           |           |       |       |                      |          |
| f_occasp                     | .2471578  | .1024504  | 2.41  | 0.016 | .0463588             | .4479568 |
| r_intel                      | .3271847  | .0407973  | 8.02  | 0.000 | .2472234             | .4071459 |
| r_ses                        | .1635056  | .0380582  | 4.30  | 0.000 | .0889129             | .2380984 |
| f_ses                        | .088364   | .0427106  | 2.07  | 0.039 | .0046529             | .1720752 |
| <b>f_occasp</b>              |           |           |       |       |                      |          |
| r_occasp                     | .2471578  | .1024504  | 2.41  | 0.016 | .0463588             | .4479568 |
| r_ses                        | .088364   | .0427106  | 2.07  | 0.039 | .0046529             | .1720752 |
| f_ses                        | .1635056  | .0380582  | 4.30  | 0.000 | .0889129             | .2380984 |
| f_intel                      | .3271847  | .0407973  | 8.02  | 0.000 | .2472234             | .4071459 |
| var(e.r_oc~p)                | .6884513  | .0538641  |       |       | .5905757             | .8025477 |
| var(e.f_oc~p)                | .6364713  | .0496867  |       |       | .5461715             | .7417005 |
| cov(e.r_oc~p,<br>e.f_occasp) | -.1582175 | .1410111  | -1.12 | 0.262 | -.4345942            | .1181592 |

LR test of model vs. saturated: chi2(4) = 1.64, Prob > chi2 = 0.8010

## Using estat scoretests to test whether constraints can be relaxed

```
. estat scoretests
(no score tests to report; all chi2 values less than 3.841458820694123)
```

No tests were reported because no tests were individually significant at the 5% level. We can obtain all the individual tests by adding the `minchi2(0)` option, which we can abbreviate to `min(0)`:

```
. estat scoretests, min(0)
Score tests for linear constraints
( 1) [r_occasp]f_occasp - [f_occasp]r_occasp = 0
( 2) [r_occasp]r_intel - [f_occasp]f_intel = 0
( 3) [r_occasp]r_ses - [f_occasp]f_ses = 0
( 4) [r_occasp]f_ses - [f_occasp]r_ses = 0
```

|      | chi2  | df | P>chi2 |
|------|-------|----|--------|
| ( 1) | 0.014 | 1  | 0.91   |
| ( 2) | 1.225 | 1  | 0.27   |
| ( 3) | 0.055 | 1  | 0.81   |
| ( 4) | 0.136 | 1  | 0.71   |

Notes:

1. When we began this example, we used `test` to evaluate potential constraints that we were considering. We obtained an overall  $\chi^2(4)$  statistic of 1.61 and thus could not reject the constraints at any reasonable level.
2. We then refit the model with those constraints.
3. For pedantic reasons, now we use `estat scoretests` to evaluate relaxing constraints included in the model. `estat scoretests` does not report a joint test. You cannot sum the  $\chi^2$  values to obtain a joint test statistic. Thus we learn only that the individual constraints should not be relaxed at reasonable confidence levels.
4. Thus when evaluating multiple constraints, it is better to fit the model without the constraints and use `test` to evaluate them jointly.

## Also see

[SEM] [example 7](#) — Nonrecursive structural model

[SEM] [sem](#) — Structural equation model estimation command

[SEM] [sem and gsem path notation](#) — Command syntax for path diagrams

[SEM] [test](#) — Wald test of linear hypotheses

[SEM] [estat scoretests](#) — Score tests