Title

stata.com

sem and gsem option constraints() — Specifying constraints

Description Syntax Remarks and examples Also see

Description

Constraints refer to constraints to be imposed on the estimated parameters of a model. These constraints usually come in one of three forms:

1. Constraints that a parameter such as a path coefficient or variance is equal to a fixed value such as 1.
2. Constraints that two or more parameters are equal.
3. Constraints that two or more parameters are related by a linear equation.

It is usually easier to specify constraints with `sem`'s and `gsem`'s path notation; see [SEM] **sem and gsem path notation**.

`sem`'s and `gsem`'s `constraints()` option provides an alternative way of specifying constraints.

Syntax

```
sem ... [, ... constraints(# [# ...]) ...]
gsem ... [, ... constraints(# [# ...]) ...]
```

where # are constraint numbers. Constraints are defined by the `constraint` command; see [R] **constraint**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Also see [R] **constraint**.

Use with sem

There is only one case where `constraints()` might be easier to use with `sem` instead of specifying constraints in the path notation. You wish to specify that two or more parameters are related and then decide you would like to fix the value at which they are related.

For example, if you wanted to specify that parameters are equal, you could type

```
. sem ... (y1 <- x@c1) (y2 <- x@c1)      (y3 <- x@c1)       ...
```

Using the path notation, you can specify more general relationships, too, such as

```
. sem ... (y1 <- x@c1) (y2 <- x@(2*c1)) (y3 <- x@(3*c1+1)) ...
```

Say you now decide you want to fix *c1* at value 1. Using the path notation, you modify what you previously typed:

```
. sem ... (y1 <- x@1) (y2 <- x@2)     (y3 <- x@4)       ...
```

Alternatively, you could do the following:

```
. constraint 1 _b[y2:x] = 2*_b[y1:x]
. constraint 2 _b[y3:x] = 3*_b[y1:x] + 1
. sem ..., ... constraints(1 2)
. constraint 3 _b[y1:x] = 1
. sem .., ... constraints(1 2 3)
```

Use with gsem

Gamma regression can produce exponential regression estimates if you constrain the log of the scale parameter to 0. Parameters associated with particular generalized linear families, such as scalar parameters, cutpoints, and the like, cannot be constrained using the `@` notation in paths. You must use Stata's constraints.

Say we wish to fit the model `y <- x1` with exponential regression. We admit that we do not remember the name under which `gsem` stores the scalar parameter, so first we type

```
. gsem (y <- x1, gamma), noestimate
```

From the output, we quickly discover that the log of the scale parameter is stored as `_b[/y:logs]`. With that information, to obtain the constrained results, we type

```
. constraint 1 _b[/y:logs] = 0
. gsem (y <- x1, gamma), constraints(1)
```

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

[R] **constraint** — Define and list constraints