Title

stata.com

Triangularize — How to triangularize a system of equations

Description Remarks and examples Also see

Description

ERMs allow endogenous covariates, but they must form a triangular system, also known as a recursive system. Said differently, ERMs do not allow simultaneous causation. This was explained for simple cases in [ERM] **Intro 3**. How to triangularize complicated systems is described below.

Remarks and examples

stata.com

The day will come when you try to fit a model and the ERM command responds with the following error:

```
. eregress y w1 w2 w3 x1 x2,
>          endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)
>          endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)
>          endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain)
>          endogenous(z1 =     z5    x1 x2 x4, nomain)
endogenous variables do not form a triangular system
   The problem may be fixable.  See triangularizing the system.
r(459);
```

The error can even occur in simple models:

```
. eregress y w1 x2 x3, endogenous(w1 = y z1 x2, nomain)
endogenous variables do not form a triangular system
    The problem may be fixable.  See triangularizing the system.
r(459);
```

The error message says the problem may be fixable. We explain below how to find the problem, how to determine whether it is fixable, and how to fix it when it is.

Remarks are presented under the following headings:

What is a triangular system?

ERMs require that the endogenous variables in the model being fit form a triangular system. The endogenous variables include the dependent variable in the main equation and the dependent variables in the `endogenous()` options. In addition, the options `entreat()`, `select()`, and `tobitselect()` add endogenous variables, but we will cover those options later.

The endogenous variables are y, w1, w2, w3, and z1 in the model

```
. eregress y w1 w2 w3 x1 x2 x5,                          ///
          endogenous(w1 =     z1 z2 x1 x2 x5, nomain)  ///
          endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)  ///
          endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain)  ///
          endogenous(z1 =     z5    x1 x2 x4, nomain)
```

The system that needs to be triangular is y, w1, w2, w3, and z1. That system is

Endogenous variable	which depends on the endogenous variable(s)
y	w1 w2 w3
w1	z1
w2	w1 z1
w3	w1
z1	*(none)*

A system is triangular when the dependencies can be ordered such that each endogenous variable is already defined before it is used as an explanatory variable. The system, in order, is

Endogenous variable	which depends on the endogenous variable(s)
z1	*(none)*
w1	z1
w3	w1
w2	w1 z1
y	w1 w2 w3

The system is in order and triangular because

1. Endogenous variable z1 depends on no other endogenous variables.
2. Endogenous variable w1 depends on z1, and z1's definition has already been listed.
3. Endogenous variable w3 depends on w1, and w1's definition has already been listed.
4. Endogenous variable w2 depends on w1 and z1, and their definitions have already been listed.
5. Endogenous variable y depends on w1, w2, and w3, and their definitions have already been listed.

When the system is triangular, ERMs can fit the model.

Triangularizing nontriangular systems

Consider the model

```
. eregress y w1 w2 w3 x1 x2 x5,
>          endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)
>          endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)
>          endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain)
>          endogenous(z1 =     z5    x1 x2 x4, nomain)
 endogenous variables do not form a triangular system
     The problem may be fixable.  See triangularizing the system.
 r(459);
```

The ERM command has already told us that the system defined by this model is not triangular. Thus, if we try to order the definitions as we did above, we will not be successful. Where we run into difficulties, however, will tell us where the problem is.

The endogenous variables in this model are `y`, `w1`, `w2`, `w3`, and `z1`. Their definitions in the order in which they appear in the command are

Endogenous variable	which depends on the endogenous variable(s)
`y`	`w1 w2 w3`
`w1`	`w2 z1`
`w2`	`w1 z1`
`w3`	`w1`
`z1`	*(none)*

The definitions in as near to the correct order as we can get them are

Endogenous variable	which depends on the endogenous variable(s)
`z1`	*(none)*
`w1`	`z1 w2` ← problem here
`w2`	`w1 z1`
`w3`	`w1`
`y`	`w1 w2 w3`

The problem appears in the second line where `w1` is defined in terms of `z1` and `w2`: `w2` has not yet been defined. Obviously, we need to put its definition above that for `w1`. However, if we move the definition of `w2` above that of `w1`, we still have a problem: `w2` depends on `z1` and `w1`, and now `w1` has not yet been defined!

You might notice that there are three endogenous variables involved in the problem—`w1`, `w2`, and `w3`—but just focus on the first pair of definitions that cause the problem. It does not matter which two of the three they are. In our case, they are

```
endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)
endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)
```

As we said in [ERM] **Intro 3**, there is a workaround for the problem when both equations are linear, as they are in this case. The workaround is

> When the simultaneous-causation problem occurs in linear equations defined by `endogenous()` options, remove the endogenous variable from one equation and substitute for it all the variables from the removed variable's equation except, of course, the variable you just removed.

The workaround in this case either

1. Removes `w2` from the first equation and substitutes "`z1 z3 x1 x2 x5`" for it.
2. Removes `w1` from the second equation and substitutes "`z1 z2 x1 x2 x5`" for it.

It does not matter which we do.

To remind you, we are fixing the first equation:

```
endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)
```

When we remove `w2` and substitute "`z1 z3 x1 x2 x5`", we obtain

```
z1 z3 x1 x2 x5   z1 z2 x1 x2 x5
```

Now, we need to remove the duplicates. Removing them, we have

```
z3            z1 z2 x1 x2 x5
```

Thus, the first equation becomes

```
endogenous(w1 = z3 z1 z2 x1 x2 x5, nomain)
```

We can now try fitting the model again:

```
. eregress y w1 w2 w3 x1 x2 x5,                              ///
            endogenous(w1 =     z3 z1 z2 x1 x2 x5, nomain)  ///
            endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)     ///
            endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain)     ///
            endogenous(z1 =     z5    x1 x2 x4, nomain)
```

When we try to fit the model, it will be successful or it will repeat the same error we saw earlier:

```
endogenous variables do not form a triangular system
    The problem may be fixable.  See triangularizing the system.
r(459);
```

In this case, the model will be successfully fit. If you do get the error, repeat the process. Remove the problems one at a time.

You can only triangularize linear equations

The rule is

> When the simultaneous-causation problem occurs in *linear* equations defined by `endogenous()` options, remove the endogenous variable from one equation and substitute for it all the variables from the removed variable's equation except, of course, the variable you just removed.

Triangularization involves a pair of equations that must both be linear. In the example above, both were linear:

```
endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)
endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)
```

They would not have both been linear if either had been fit by `probit` or `oprobit`. If one or both of the equations had been

```
endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain probit)
endogenous(w2 = w1  z1 z3 x1 x2 x5)
```

or

```
endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)
endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain oprobit)
```

there would have been no solving the simultaneous-causation problem.

This linearity requirement applies only to the two equations directly involved. Other equations can be nonlinear and there will be no issue. The workaround we outlined would have worked just as well had the model been

```
. eregress y w1 w2 w3 x1 x2 x5,                                    ///
            endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)            ///
            endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)            ///
            endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain probit)     ///
            endogenous(z1 =     z5    x1 x2 x4, nomain probit)
```

or even

```
. eprobit y w1 w2 w3 x1 x2 x5,                                     ///
            endogenous(w1 = w2  z1 z2 x1 x2 x5, nomain)            ///
            endogenous(w2 = w1  z1 z3 x1 x2 x5, nomain)            ///
            endogenous(w3 = w1  z4 z5 x1 x2 x5, nomain probit)     ///
            endogenous(z1 =     z5    x1 x2 x4, nomain probit)
```

Options entreat(), select(), and tobitselect() also add endogenous variables

The example above contained repeated uses of the `endogenous()` option. When you make the list of endogenous variables, you must also include the dependent variables *treated* and *selected* from the options

`entreat(`*treated*`= ...)`

`select(`*selected*`= ...)`

`tobitselect(`*selected*`= ...)`

The above options make *treated* and *selected* endogenous. Unlike with the `endogenous()` option, however, the variables are not automatically added to the main equation even if you do not specify `nomain`.

These three options are nonlinear. If the simultaneous-causation problem involves equations created by these options, then there is no workaround for the simultaneous-causation problem.

Workarounds involving the main equation

The example of the simultaneous-causation problem involved two equations defined by `endogenous()` options. The problem could also occur when one of the equations is the main equation. In [ERM] **Intro 3**, we discussed problems involving the main equation as if they were different from simultaneous causation, but they are not. It is the same problem that has the same workaround, but with an important difference.

In workarounds involving equations defined by `endogenous()` equations, the workaround may be applied to either equation.

In workarounds involving the main equation and an `endogenous()` equation, the workaround must be applied to the `endogenous()` equation.

> When the simultaneous-causation problem involves the main equation fit by `eregress` and an `endogenous()` linear equation, remove the dependent variable from the `endogenous()` equation and substitute for it all the variables from the main equation except, of course, the variable you just removed.

Also notice that this rule applies only to main equations fit by `eregress`. What about `eintreg`, `eprobit`, and `eoprobit`?

The simultaneous-causation problem does not arise in models fit by `eintreg`. There is no way you could include `eintreg`'s dependent variables as explanatory variables in another equation.

The simultaneous-causation problem can arise in models fit by `eprobit` and `eoprobit`, but those are nonlinear equations, and that means you cannot apply the workaround. The workaround requires that both equations be linear.

The main equation must be linear if it is one of the two equations involved in the simultaneous-causation problem. Otherwise, the main equation is not required to be linear.

Why the above is a workaround and not a fix

It is a detail, but you may have noticed that we provided a workaround and not a fix. The purpose of ERMs is to obtain valid estimates of the coefficients of the main equation—its structural parameters—in light of lots of complications. It so happens that ERMs produce estimates of structural parameters for all the other equations if the system is truly triangular. That is not important, but it is true.

When you triangularize a nontriangular system, ERMs no longer produce estimates of the structural parameters for the equations that you modify. They produce estimates of the reduced-form equation, and that is sufficient. Valid estimates of the reduced-form equation ensures that estimates of the coefficients in the main equation are estimates of its structural parameters.

Thus, what we provided is a workaround, not a fix. If you use the workaround, do not interpret any equations modified as estimates of their structural parameters.

Also see

[ERM] **Intro 3** — Endogenous covariates features