The Stata Journal (2004)
4, Number 4, pp. 442–448

From the help desk: Seemingly unrelated regression with unbalanced equations

Allen McDowell
StataCorp

Abstract. This article demonstrates how to estimate the parameters of a system of seemingly unrelated regressions when the equations are unbalanced, i.e., when the equations have an unequal number of observations. With estimators that require the data to be in wide format, such as Stata's `sureg`, the equations must be balanced. Any additional observations that are available for some equations, but not for all, are discarded, potentially resulting in a loss of efficiency. Reshaping and scaling the data allows us to use Stata's `xtgee` command to fit the model and obtain estimates utilizing all the available data. The resulting estimator is potentially more efficient when the equations are unbalanced.

Keywords: st0079, SUR, seemingly unrelated regression, unbalanced equations, generalized estimating equations

1 Introduction

Fitting a system of equations via seemingly unrelated regression (SUR) with Stata's `sureg` or `reg3` commands will result in a loss of information if the number of observations is not the same for all equations. The `xtgee` command provides an alternative estimator that can use all the available information, and for normally distributed data, `xtgee`'s iteratively reweighted least-squares estimator is equivalent to maximum likelihood.

2 Preparing the data

To help you visualize the steps needed to prepare your data, note that a system of three seemingly unrelated equations that can be written as

$$\begin{aligned}\mathbf{y}_1 &= \mathbf{X}_1\boldsymbol{\beta}_1 + \boldsymbol{\epsilon}_1\\ \mathbf{y}_2 &= \mathbf{X}_2\boldsymbol{\beta}_2 + \boldsymbol{\epsilon}_2\\ \mathbf{y}_3 &= \mathbf{X}_3\boldsymbol{\beta}_3 + \boldsymbol{\epsilon}_3\end{aligned}$$

can also be written as one superequation

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\end{bmatrix} = \begin{bmatrix}\mathbf{X}_1 & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{X}_2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{X}_3\end{bmatrix}\begin{bmatrix}\boldsymbol{\beta}_1\\ \boldsymbol{\beta}_2\\ \boldsymbol{\beta}_3\end{bmatrix} + \begin{bmatrix}\boldsymbol{\epsilon}_1\\ \boldsymbol{\epsilon}_2\\ \boldsymbol{\epsilon}_3\end{bmatrix}$$

st0079

This rearrangement of multiple linear equations into a single superequation generalizes for any number of equations. The key point of interest here is the block diagonal arrangement of the $\mathbf{X}_i$ and the fact that the off-diagonal blocks of the data matrix are populated with zeros. Assuming that you have a dataset in wide form, which is the most natural form for multiple-equation data, you need to take three steps to prepare the data before estimating with `xtgee`:

1. scale the data to account for equation-specific variances
2. reshape the data into long form
3. `tsset` the data

Let's begin by simulating some data. First, we generate a set of normally distributed, correlated error terms for the three equations:

```
. set seed 2004
. mat c1 = (1, .9, .6 \ .9, 1, .75 \ .6, .75, 1)
. drawnorm e1 e2 e3, n(100) mean(0 0 0) sds(10 15 20) corr(c1)
(obs 100)
```

Next we generate a set of covariates:

```
. mat c2 = (1,-.6,-.009,.49,-.38,.002\-.6,1,-.59,-.608,-.08,-.338\-.009,
>  -.59,1,-.18,-.11,.144\.49,-.608,-.18,1,.46,.18\-.38,-.08,-.11,.46,1,
> .004\.002,-.338,.144,.18,.004,1)
. drawnorm x11 x12 x21 x22 x31 x32, mean(10 5 19 20 15 12) sds(13 20 27 2 8 22)
>  corr(c2)
```

We can now generate the dependent variables:

```
. generate y1 = 100 + 15*x11 + .7*x12 + e1
. generate y2 = 75 + 25*x21 + 20*x22 + e2
. generate y3 = 50 + 15*x31 + 19*x32 + e3
```

Finally, we set a few observations to missing so that the equations are unbalanced and generate a time variable to index the observations. The time variable will be used as the panel identifier when we `tsset` the data since it indexes contemporaneous observations across equations.

```
. replace y1 = . in 1/3
(3 real changes made, 3 to missing)
. replace y2 = . in 97/100
(4 real changes made, 4 to missing)
. replace x31 = . in 25
(1 real change made, 1 to missing)
. gen time = _n
. save wide, replace
file wide.dta saved
```

Fitting a SUR model with `sureg` we get

```
. sureg (y1 x11 x12) (y2 x21 x22) (y3 x31 x32)
Seemingly unrelated regression
----------------------------------------------------------------------
Equation          Obs  Parms        RMSE    "R-sq"       chi2        P
----------------------------------------------------------------------
y1                 92      2     9.16867    0.9973  123197.83   0.0000
y2                 92      2    14.00425    0.9996  829710.24   0.0000
y3                 92      2    17.88595    0.9978   77775.54   0.0000
----------------------------------------------------------------------

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
y1           |
         x11 |   15.01983   .0517867   290.03   0.000     14.91833    15.12133
         x12 |   .7003301   .0466798    15.00   0.000     .6088394    .7918207
       _cons |   98.10696   1.233664    79.52   0.000     95.68902    100.5249
-------------+----------------------------------------------------------------
y2           |
         x21 |   24.99282   .0330073   757.19   0.000     24.92812    25.05751
         x22 |    19.7629   .4412294    44.79   0.000     18.89811    20.62769
       _cons |   77.81727   9.237519     8.42   0.000     59.71207    95.92248
-------------+----------------------------------------------------------------
y3           |
         x31 |   14.79411   .1726112    85.71   0.000      14.4558    15.13242
         x32 |   19.00814    .071608   265.45   0.000      18.8678    19.14849
       _cons |   50.27515   3.244142    15.50   0.000     43.91675    56.63355
------------------------------------------------------------------------------
```

Notice that all three equations have only 92 observations in the estimation sample, so we lost a total of 24 observations. We lost three observations in each equation because y_1 was missing in the first three observations; we lost four observations in each equation because y_2 was missing in the last four observations, and we lost one observation in each equation because x_{31} was missing in the 25th observation. By rearranging our data into long form and fitting a single-equation panel-data model, we will be able to recover 16 of the missing observations, restricting the loss of information due to missing values to the specific equations from which the data are actually unobserved.

Before we can fit the same model using `xtgee`, we must first rescale the data. While `sureg` allows for unequal error variances across equations, `xtgee` assumes that errors are homoskedastic. To allow for unequal error variances, we can fit separate OLS regressions for each equation and rescale the variables for each equation using their respective regression root mean squared error.

```
. quietly regress y1 x11 x12
. foreach v of var y1 x11 x12 {
  2.        replace `v' = `v' / e(rmse)
  3. }
(97 real changes made)
(100 real changes made)
(100 real changes made)
. generate cons1 = 1/e(rmse)
. quietly  regress y2 x21 x22
. foreach v of var y2 x21 x22 {
  2.         replace ‘v’ = ‘v’ / e(rmse)
  3. }
(96 real changes made)
(100 real changes made)
(100 real changes made)
. generate cons2 = 1/e(rmse)
. quietly regress y3 x31 x32
. foreach v of var y3 x31 x32 {
  2.         replace ‘v’ = ‘v’ / e(rmse)
  3. }
(100 real changes made)
(99 real changes made)
(100 real changes made)
. generate cons3 = 1/e(rmse)
. save rescaled, replace
file rescaled.dta saved
```

Note that a scaled constant was generated for each equation. These scaled constants will be used instead of a single intercept in the xtgee version of the model.

We must now save the data for each equation in a separate dataset, perform some data manipulations, and append the three datasets together. However, before we proceed, a short digression on the mechanics of appending datasets is called for; the reason will become readily apparent as we progress.

3 A digression on appending datasets

Suppose that you have two datasets, A and B. In dataset A, you have two variables, var1 and var2; in dataset B, you have two variables, var1 and var3. For simplicity, suppose that there are just two observations in each dataset.

```
. use A
. list

     +-------------+
     | var1   var2 |
     |-------------|
  1. |    1      2 |
  2. |    .      2 |
     +-------------+

. use B, clear
. list

     +-------------+
     | var1   var3 |
     |-------------|
  1. |    1      3 |
  2. |    1      3 |
     +-------------+
```

It is important that you know what to expect when we append these two datasets together.

```
. use A, clear
. append using B
. list

     ┌────────────────────┐
     │ var1   var2   var3 │
     ├────────────────────┤
  1. │    1      2      . │
  2. │    .      2      . │
  3. │    1      .      3 │
  4. │    1      .      3 │
     └────────────────────┘
```

Notice the missing values that are generated. In the case of `var1`, which was common to both datasets, the observations from `B` were stacked under the observations from `A`. The missing value that was present in `A` is still present in the appended dataset. Now look at `var2` and `var3` in the appended dataset. `var2` was unique to `A`, and when the observations from `B` were appended, missing values were generated for `var2` in the observations that originated in `B`. Similarly, `var3` was unique to `B`, and when the observations from `B` were appended, missing values were generated for `var3` in the observations from `A`. Before we can fit the model with `xtgee`, we must be able to distinguish between the missing values, such as those for `var1` in the appended dataset, that are missing because the data were actually missing in one of the original datasets and those missing values that were generated for `var2` and `var3` because the respective variables were absent from one of the original datasets. We will want to recode the missing values of the latter type to zeros, leaving the missing values of the former type as missing in the appended dataset.

4 Preparing the data, continued

After saving the data from each equation in a separate dataset, we will `rename` the dependent variables so that they share a common variable name. We will then `append` all the datasets together. When we do so, blocks of missing values will be generated, just as demonstrated above. These missing values will then be recoded to zeros, thus forming the off-diagonal blocks of zeros that were described above when we converted the multiple equations into a single superequation. As discussed in *A digression on appending datasets*, when recoding the missing values that are generated by appending the datasets together, we must take care not to recode the missing values that were present in the original dataset. As we break up the original wide dataset into individual equation-specific datasets, we will generate variables that indicate if an observation has any missing values. These variables should share a common name in all the equation-specific datasets so that, when they are appended together, we have a single variable that marks the observations for which there were missing values to be retained in the appended dataset. We also need to generate a new variable for each dataset that identifies the equation from which the data originates; these variables should also share a common name so that, when the datasets are appended, we have a single variable that identifies the equation of origin for each observation. This new identification variable, along with the time variable from the original dataset, will allow us to properly `tsset` the data for `xtgee`.

```
. preserve
. keep y1 x11 x12 cons1 time
. mark sample
. markout sample y1 x11 x12 cons1
. rename y1 y
. gen id = 1
. save data1, replace
file data1.dta saved
. restore
. preserve
. keep y2 x21 x22 cons2 time
. mark sample
. markout sample y2 x21 x22 cons2
. rename y2 y
. gen id = 2
. save data2, replace
file data2.dta saved
. restore
. preserve
. keep y3 x31 x32 cons3 time
. mark sample
. markout sample y3 x31 x32 cons3
. rename y3 y
. gen id = 3
. save data3, replace
file data3.dta saved
. restore
. clear
. use data1
. append using data2
. append using data3
. mvencode x* cons* if sample, mv(0)
         x11: 195 missing values recoded
         x12: 195 missing values recoded
         x21: 196 missing values recoded
         x22: 196 missing values recoded
         x31: 193 missing values recoded
         x32: 193 missing values recoded
       cons1: 195 missing values recoded
       cons2: 196 missing values recoded
       cons3: 193 missing values recoded
. tsset time id
       panel variable:  time, 1 to 100
        time variable:  id, 1 to 3
```

Note that, when the data is `tsset`, the panel-identification variable and the time variable have reversed the roles that they usually perform. Since we are interested in modeling the contemporaneous correlation across equations, time is the relevant panel identifier, and the equation identifier indexes the repeated measures within panel.

5 Fitting the SUR model with xtgee

We are now ready to fit the SUR model using `xtgee`. The appropriate specification requires that we fit the model with a Gaussian family, an identity link, and since the SUR model imposes no structure on the correlation matrix, an unstructured within-group correlation structure. Since we also have generated rescaled constants for each equation, we must explicitly include them in our model and specify the `noconstant` option.

```
. xtgee y x* cons*, family(gaussian) link(identity) corr(unstructured) noconstant
Iteration 1: tolerance = .11677885
Iteration 2: tolerance = .00171051
Iteration 3: tolerance = 5.458e-06
Iteration 4: tolerance = 3.981e-07

GEE population-averaged model                   Number of obs      =       292
Group and time vars:                time id     Number of groups   =       100
Link:                              identity     Obs per group: min =         2
Family:                            Gaussian                    avg =       2.9
Correlation:                   unstructured                    max =         3
                                                Wald chi2(8)       =   2021334
Scale parameter:                   .9770902     Prob > chi2        =    0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         x11 |   15.00825   .0576632   260.27   0.000     14.89523    15.12127
         x12 |   .6771566   .0474135    14.28   0.000     .5842279    .7700853
         x21 |   25.00623    .032086   779.35   0.000     24.94335    25.06912
         x22 |   19.88076   .4328133    45.93   0.000     19.03246    20.72906
         x31 |   14.75204   .1648838    89.47   0.000     14.42887    15.07521
         x32 |   19.01571    .066133   287.54   0.000     18.88609    19.14533
       cons1 |   98.25454   1.277547    76.91   0.000     95.75059    100.7585
       cons2 |   75.11264    9.01052     8.34   0.000     57.45235    92.77294
       cons3 |   50.68025   3.110645    16.29   0.000      44.5835    56.77701
------------------------------------------------------------------------------
```

Inspection of the output indicates that we have accomplished our goal. Rather than losing 24 observations due to missing values, the `xtgee` estimator was able to fit the model losing only the 8 observations for which the data were actually unobserved.

About the Author

Allen McDowell is Director of Technical Services at StataCorp.