Lab 2: OLS regression

Andreas Beger

February 2, 2009

1 Overview

This lab covers basic OLS regression in Stata, including:

- multivariate OLS regression
- reporting coefficients with different confidence intervals
- reporting standardized regression coefficients
- calculating predicted values of Y
- calculating residual values
- plotting residuals against an independent variable
- F-tests for joint significance of independent variables

The dataset I use as an example comes from the replication data for Lacina (2006). The paper examines the number of fatalities in civil wars. Each observation in the dataset is a particular civil war, and the data include 112 civil wars between 1946 and 2002. The dependent variable is going to be `battledeadbest`, a variable that captures the number of battle deaths in a particular civil war. Once you open the dataset in Stata, you can type `describe` to get a list of variables and their labels. Type `summarize` to get a list of summary statistics for each variable.

2 Multivariate regression

Stata has a large number of regression commands, but they all share the same syntax. Following the particular regression command (e.g. `regression` or `logit`) is whatever variable corresponds to the dependent variable, as well as a list of variable that you want to use as independent variables in your model.[1] Stata will by default include a constant term in most regressions, so the list of independent variables is actually optional. All regression commands in Stata also have various options that allow you to change some of the things Stata assumes by default, or to

[1] Some regression commands do not allow you to specify a dependent variable because it is already implied by the data you have, e.g. with time-series data.

change various other aspects of the model. If you want to specify an option, type a comma after the list of variables, followed by the options. The help file for each regression command lists the options available for that model.

In our case, I want to estimate an OLS regression model in which the number of battle deaths is the dependent variable, and the independent variables are the natural log of population, number of military personnel, military expenditures, and three dummy variables indicating a secessionist conflict, a civil war with foreign intervention, and a state that is dominated by a particular ethnic group at the expense of other ethnic groups:

```
. reg battledeadbest lnpop milper milex secession intervention ethnicpolar

      Source |       SS       df       MS              Number of obs =     112
-------------+------------------------------           F(  6,   105) =    3.29
       Model |  9.5174e+11     6  1.5862e+11           Prob > F      =  0.0053
    Residual |  5.0662e+12   105  4.8250e+10           R-squared     =  0.1582
-------------+------------------------------           Adj R-squared =  0.1100
       Total |  6.0179e+12   111  5.4216e+10           Root MSE      =  2.2e+05

------------------------------------------------------------------------------
battledead~t |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       lnpop |   24633.36   18441.92     1.34   0.185    -11933.57    61200.29
      milper |   35.72413   36.66746     0.97   0.332    -36.98066    108.4289
       milex |  -.0026404   .0036212    -0.73   0.468    -.0098205    .0045397
   secession |   -96164.1   50556.39    -1.90   0.060    -196408.1    4079.873
intervention |   102344.4   45396.46     2.25   0.026     12331.65    192357.2
 ethnicpolar |  -124928.2   55828.77    -2.24   0.027    -235626.4   -14230.11
       _cons |  -279473.4   301128.2    -0.93   0.355    -876554.9    317608.1
------------------------------------------------------------------------------
```

As you can see, Stata made a few choices by default, like choosing to report a 95% confidence interval for the coefficient estimates. If you look at the help file for `regress` you can see that there are a few options that allow you to change these default choices. For example, if I was interested in 90% confidence intervals for the coefficient estimates rather than 95% confidence intervals, I could have added "`, level(90)`" after the regression command above:

```
. reg battledeadbest lnpop milper milex secession intervention ethnicpolar, l(90)

      Source |       SS       df       MS              Number of obs =     112
-------------+------------------------------           F(  6,   105) =    3.29
       Model |  9.5174e+11     6  1.5862e+11           Prob > F      =  0.0053
    Residual |  5.0662e+12   105  4.8250e+10           R-squared     =  0.1582
-------------+------------------------------           Adj R-squared =  0.1100
       Total |  6.0179e+12   111  5.4216e+10           Root MSE      =  2.2e+05

------------------------------------------------------------------------------
battledead~t |      Coef.   Std. Err.      t    P>|t|     [90% Conf. Interval]
-------------+----------------------------------------------------------------
       lnpop |   24633.36   18441.92     1.34   0.185    -5970.929    55237.65
      milper |   35.72413   36.66746     0.97   0.332    -25.12534     96.5736
       milex |  -.0026404   .0036212    -0.73   0.468    -.0086497    .0033689
   secession |   -96164.1   50556.39    -1.90   0.060    -180062.2   -12266.01
intervention |   102344.4   45396.46     2.25   0.026     27009.23    177679.7
 ethnicpolar |  -124928.2   55828.77    -2.24   0.027    -217575.8   -32280.66
       _cons |  -279473.4   301128.2    -0.93   0.355    -779194.2    220247.4
------------------------------------------------------------------------------
```

Or, if you want standardized regression coefficients to be reported, you can add the `beta` option to the regression command:

```
. reg battledeadbest lnpop milper milex secession intervention ethnicpolar, beta

      Source |       SS       df       MS              Number of obs =     112
-------------+------------------------------           F(  6,   105) =    3.29
       Model |  9.5174e+11     6  1.5862e+11           Prob > F      =  0.0053
    Residual |  5.0662e+12   105  4.8250e+10           R-squared     =  0.1582
-------------+------------------------------           Adj R-squared =  0.1100
       Total |  6.0179e+12   111  5.4216e+10           Root MSE      =  2.2e+05

------------------------------------------------------------------------------
battledead~t |      Coef.   Std. Err.      t    P>|t|                     Beta
-------------+----------------------------------------------------------------
       lnpop |   24633.36   18441.92     1.34   0.185                  .183521
      milper |   35.72413   36.66746     0.97   0.332                 .1244649
       milex |  -.0026404   .0036212    -0.73   0.468                -.0719571
   secession |   -96164.1   50556.39    -1.90   0.060                -.2017616
intervention |   102344.4   45396.46     2.25   0.026                 .2194886
 ethnicpolar |  -124928.2   55828.77    -2.24   0.027                 -.217714
       _cons |  -279473.4   301128.2    -0.93   0.355                        .
------------------------------------------------------------------------------
```

3 Post-estimation commands

Usually there are various things you might want to do after estimating a regression model in Stata, like calculating predicted values, calculating residual values, checking for heteroscedasticity, etc. You could do these by hand, but Stata conveniently provides various post-estimation commands that automate some of these common tasks. If you look at the help file for an estimation command in Stata, like `regress`, there usually will be a link at the top right of the help file that directs you to a list of post-estimation commands available for that estimation command. Alternatively, you could also type `help regress postestimation` to get to the same help file.

3.1 Predicted values

In this case, I want to calculate predicted values of Y, as well as the residual values for each case. To do that I can use the `predict` command and the appropriate option. The `predict` command generates a new variable that contains whatever you asked it to calculate. To get predicted values of Y, I could type `predict yhat, xb`. This will generate a new variable named `yhat` that contains the linear prediction for each case in my dataset, using the independent variable values in that case as well as the coefficient estimates from the last regression that was estimated in Stata. There are two important points with this command: (1) it uses the coefficient estimates from the last regression that was estimated in Stata, so be sure it is the right one, and (2) the command often calculates a linear prediction by default, but sometimes that is not what you will want to calculate.

To calculate residual values I can use the same command with a different option, e.g. `predict e, residual`.

After you have calculated predicted values or residuals, you can use the `list` command to look through predicted values for each case, or you can browse the dataset itself. They are stored just like any other variable in the dataset, so the same commands apply to them, like `summarize`.

3.2 Plotting residuals

After calculating residual values, you might be interested in examining them in more detail to see if you violate any of the assumptions of OLS regression. One way of doing this is to visually examine plots of the residual values against an independent variable. There are two ways of doing this in Stata.

The first, and faster way, is to use the `plot` command. To produce a plot of residuals against the natural log of population, type `plot e lnpop`, which gives the following output:

```
. plot e lnpop

 1.9e+06 +
         |                                *
         |
         |
         |
         |
    R    |
    e    |
    s    |
    i    |
    d    |
    u    |
    a    |                                                            *
    l    |
    s    |                             *
         |
         |                 *              *
         | *            *   *   *  *      **              *
         | *    *     ****  * **** ********* ** *****    ** **     * *  ** *
         |       * *  *** **  ***    *  **  ** * * ** **  *               *
 -265722 +  *                       *   **     *                      ***
          +----------------------------------------------------------------+
          13.3229                     Ln pop                        20.65
```

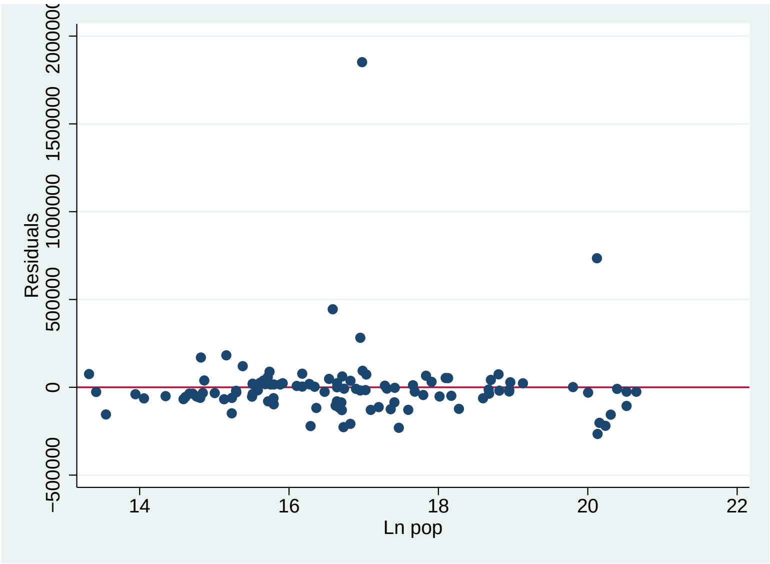
That is obviously not very pretty, and unless you routinely copy and paste from you log files to papers you are writing, also not useful in the long run. But it is very fast and easy.

The second, and harder, way is to use Stata's normal graph commands. This produces much nicer results, but can also take some time since graphing commands take longer to execute and are more complicated. To get the equivalent scatter plot from above, type `graph twoway scatter e lnpop`.

3.3 Joint significance tests

Another set of post-estimation commands allows you to conduct various statistical tests relating to your coefficient estimates or model specification. For example, if I wanted to conduct an

Figure 1: Plot of residuals against population



F-test with the null hypothesis that the coefficient estimates for `milper` (Number of military personnel) and `milex` (Military expenditures) are jointly equal to zero, I could type:

```
. test milper milex

 ( 1)  milper = 0
 ( 2)  milex = 0

       F(  2,   105) =    0.71
            Prob > F =    0.4963
```

The commands `test` and `testparm` also allow you to test more complicated hypotheses. The help files have specific examples.

4 Brief aside on Stata macros

When you estimate regression models in Stata you will get the familiar output table, but Stata also stores all the information in those tables internally in a way that is accessible to you if you know where to look. Type `ereturn list` after a regression command to see everything Stata has stored. This can be helpful if you want to manipulate some part of a regression output after running the regression.

```
. ereturn list

scalars:
                  e(N) =  112
               e(df_m) =  6
               e(df_r) =  105
                  e(F) =  3.287564493621457
```

```
                 e(r2) =  .158150537290226
               e(rmse) =  219657.8626149909
                e(mss) =  951741569343.9385
                e(rss) =  5066205543901.552
               e(r2_a) =  .1100448537068104
                 e(ll) =  -1532.887520360734
               e(ll_0) =  -1542.528148062565

macros:
            e(cmdline) : "regress battledeadbest lnpop milper milex secession intervention ethnicpolar"
              e(title) : "Linear regression"
                e(vce) : "ols"
             e(depvar) : "battledeadbest"
                e(cmd) : "regress"
         e(properties) : "b V"
            e(predict) : "regres_p"
              e(model) : "ols"
          e(estat_cmd) : "regress_estat"

matrices:
                  e(b) :  1 x 7
                  e(V) :  7 x 7

functions:
             e(sample)
```

For example, Stata stores all coefficient values in a vector called "`e(b)`". So if you wanted to calculate the predicted number of fatalities for some conflict with values for the independent variables of your choosing, you could multiply a vector of independent variable values with the vector of coefficients to get a point estimate for fatalities.

You can also reference individual coefficient values as `_b[name]` and individual standard error values as `_se[name]`, where name corresponds to the variable of interest.

```
. display _b[lnpop] _skip(2) _se[lnpop]
24633.36  18441.925
```

5 Exercise

You will basically replicate the examples I have used above. Your answer should consist of a single Stata do file that creates a log that contains all the necessary Stata output to answer the questions below.

1. Estimate an OLS regression model of battle deaths on population, military personnel, military expenditures, secession, intervention, and ethnic polarization.

2. Report 90 and 99% confidence intervals in addition to the default 95%.

3. Report standardized coefficients.

4. Calculate predicted values and residuals.

5. Get Stata to list the conflict name, battle deaths, predicted fatalities, and residual for the Sri Lankan civil war, and that conflict only (the ID for that conflict is 71...).

6. Create a graph that shows the residual values (y-axis) plotted against lnpop (x-axis). Add a horizontal line at 0. Save or export the graph.

7. Conduct a F-test of the hypothesis that lnpop, milper, and milex are jointly 0.

8. In Stata, calculate the predicted number of fatalities for a hypothetical conflict with the following values in independent variables: lnpop=17, milper=1000, milex=1000, not a secessionist conflict, there was a foreign intervention, the state is not ethnically polarized. (Hints: create a vector containing the independent variable values, including a 1 at the end for the constant term, create a vector containing the coefficient values (matrix iv = e(b)), multiply the two, and display the result. You will need the command `matrix` and `matlist`.)

References

Lacina, Bethany. 2006. "Explaining the Severity of Civil Wars." *Journal of Conflict Resolution* 50(2): 276–289.