CODERS' CORNER

CSAE
CENTRE FOR THE STUDY OF AFRICAN ECONOMIES

BOOTSTRAPPED SE WITH THREE-WAY PANELS IN STATA

Stata has excellent and flexible bootstrapping functions for cross-sectional, time series and panel data structures. However, it lacks automatic capability in dealing with three-way panel structures, that is, data structures indexed by *i, j, t*, where for every *i* there is every j and for every *i, j* there is every *t*.

Examples of this data structure include: migration flows between countries over time, social interactions between individuals over time and trade between any unit over time. In general, this structure will often appear whenever you consider multiple dimensions.

As a simplistic example, suppose we are interested in studying trade across countries – we construct a data set of trade flows from every country to every other country over a 10-year period. We are interested in the effect of free trade agreements on trade volume. However, having a free trade agreement with another country is likely to be endogenous. Suppose we have some instrument to deal with this and can proceed with an instrumental variables approach. Free trade membership is a dichotomas variable and therefore we have a probit first stage, and to avoid the "forbidden regression" problem we use a control function methodology. We have a two-step estimator, so in order to correct standard errors for first stage estimation error we can take a bootstrapping approach. In this setting, the above approach allows us to include a wide variety of flexible fixed effects in the second stage, as well as use block-bootstrapped standard errors.

To carry out the above procedure, we first define an estimation program. This program performs whatever estimation you want to bootstrap.

```
capture program drop estimation
      program define estimation, rclass
            "Estimation that you want bootstrap standard errors for"
      End
```

Next, we write a program that constructs the bootstrap sample that we wish to run estimation on in each iteration. Here the researcher must decide on which dimensions they want to draw replications from. That is, in a block-bootstrap panel procedure you draw entire panels, not observations, in order to better replicate the data generating process. Following this reasoning in the example I draw *i, j* blocks with replacement to allow for dependence within a block. Effectively I'm constructing a *(i, j), t* panel structure.

```
capture program drop sample_construction
program define sample_construction, rclass
      preserve
            capture drop sample_panel
            egen sample_panel=group(i j)
            bsample, cl(sample_panel) idcluster(new_panel_id)
            unique i
            local N=r(unique)
            bysort j t(i): replace i=i+_n+`N'

      estimation

      mat b1=e(b)
      return matrix b=b1
restore
end
```

The sample construction program will be replicated *N* times where *N* is the number of repetitions specified by the user. The preserve and restore wrappers ensure that the in-use dataset is not altered. We begin by dropping previous incarnations of *sample_panel* which specifies the dimension on which we take draws. Then we generate the new *sample_panel* variable[1] as the unique combinations of *i, j*.

Next, we call the Stata command bsample which re-samples the data (with replacement) clustering (that is blocking) at the *sample_panel* level and generates a new unique identifier for each sampled panel as *new_panel_id*. The next three lines are optional but allow retention of the three-way panel dataset structure, that is they ensure that the new dataset is uniquely identified by *i, j, t*, which may be required when running the estimation command. We then run our predefined estimation command.

After the estimation command, we specify the statistic we want this program to return each time it is called. In our case we are interested in the standard errors of the vector of coefficients b.
Finally, we write a third program that performs the simulation described by sample_panel which performs the estimation method stipulated by estimation.

```
capture program drop three_way_boot_se
     program define three_way_boot_se, rclass
          qui:estimation
          local n=e(N)
          mat b0=e(b)

          stimulate , reps(250) seed(42) : sample_construction

          bstat , stat(b0) n(`n')
end
```

This program first runs an initial call to estimation. Standard results are stored in b0 and are used for comparison. The Stata command simulate specifs the number of reps and seed. In this case we tell Stata to run the program sample_construction 250 times storing the returned result.

Finally, we use another Stata command bstat which calculates and displays results from our bootstrapped simulations. The options *stat(b0)* and *n('n')* ensure that the original standard errors (that is non-bootstrapped version) and sample size are also reported in the results table.

[1] Construction of the *sample_panel* variable need not happen within this program and a faster procedure would move its creation to the three_way_boot_se program.

Luke Milsom, DPhil Candidate in Economics, University College, Oxford
04 March 2019