PSet 5 Hints for Stata users

Raymond P. Guiteras

MIT 14.382 Spring 2006

I have tested these tips in Stata 7.0, 8.0 and 8.2. I am not sure if they will work in Stata 9.0 (the version available to everyone at MIT,), but you can always "dumb down" your code to a previous version by entering `version 8.0` as the first line of your do-file.

1. Loops

   `-forvalues-` allows you to loop through iterations of a command.

   Sample code:

   ```
   forvalues i = 1/10 {
     display "The value of the macro i is `i'"
     }
   ```

   `-forvalues-` loops can be nested as follows:

   ```
   forvalues i = 1/5 {
     forvalues j= 1/5 {
       display "The value of the macro i is `i'"
       display "The value of the macro j is `j'"
       scalar k = `i'+`j'
       display "The value of the scalar k is " k
       }
     }
   ```

   Note that macros must be called using left-single-quote and right-single-quote, but scalars are just called like a variable.

   There are other commands such as `-while-` that will allow you to do much the same thing.

2. Accessing regression output

   After running *estimation-class* commands such as `-reg-` and `-qreg-`, Stata stores lots of useful results. You can see what is stored and where by typing `ereturn list` after your `-reg-` command. For example:

   ```
   use "C:\Stata8\auto.dta", clear
   reg price mpg
   ```

ereturn list

You can see that the coefficients are stored in a 2x1 matrix called `e(b)` and the variance-covariance matrix is stored in a matrix called `e(V)`. There are other useful results too, such as the sample size, R-squared and adjusted R-squared, etc. Note that here `e(b)` is a 2x1 matrix because we are estimating two parameters – the coefficient on mpg and the intercept term. For `-qreg-`, things are pretty much the same.

To see what the matrix `e(b)` contains, enter `matrix list e(b)`.

3. Creating matrices

   To create a 10x2 matrix of missing values, enter `matrix MyTestMat = J(10,2,.)`.

   To see what MyTestMat looks like, enter `matrix list MyTestMat`

   To see what matrices are stored in memory, enter `matrix dir`

   To create a 2x10 matrix of zeros, enter `matrix MyTestMat = J(2,10,0)`.

   Suppose you've just run a regression as in the previous section. To replace the first row of MyTestMat with the coefficient vector `e(b)`, enter `matrix MyTestMat[1,1]=e(b)`. (It may look a little strange to refer to the (1,1) cell of MyTestMat when you want to replace the entire first row, but this is the right command.)

   Suppose you're running a forvalues loop to replicate a regression a number of times and using `i` as your counter macro. To replace the ith row of MyTestMat with the coefficient vector `e(b)` estimated in the ith replication, enter `matrix MyTestMat[‘i’,1]=e(b)` . Note the use of left- and right-single-quotes to call the macro i.

4. How to create a fake data set with 20 observations

   clear

   set more off

   scalar T=20

   matrix FakeData = J(T,1,.)

   svmat FakeData, names(placeholder)

   rename placeholder1 placeholder

   replace placeholder=_n

   gen x1=uniform() /*so x1~U[0,1]*/

   replace x1=(4*x1)+1 /*now x1~U[1,5]*/

   gen e=invnorm(uniform()) /*e~N(0,1) uncorrelated*/

   scalar beta = 1

   gen y=(beta*x1) + e

   save FakeData.dta, replace

5. Generating random numbers with a specified distribution.

   First strategy: transform from a uniform distribution. See Casella and Berger 5.6 and Theorem 2.1.10.

`scalar u=uniform()` draws a scalar from a U[0,1]

`generate u=uniform()` creates a variable (as many observations as are in your dataset) with each observation drawn from U[0,1] distribution, and independent.

`gen e1=invnorm(uniform())` creates a variable with each observation drawn from N[0,1] distribution, and independent.

`e2=invttail(5,uniform())` creates a variable with each observation drawn from t(5) distribution, and independent.

`e3=invttail(1,uniform())` creates a variable with each observation drawn from t(1) distribution, and independent. Why is this relevant?

`e4=invchi2(2,uniform())` creates a variable with each observation drawn from chi-squared(2) distribution, and independent.

Second strategy: find a command that gives you the distribution you want.