Problem Set 1 -Setup Computational Environment-

Yuta Toyama

Last updated: September 28, 2018
Due date: Beginning of the class on October 15

Remarks

1. Use A4 report papers, staple them in the upper left corner.
2. Type your answer.
3. You are allowed (and encouraged) to form a study group up to 3.
4. Attach the print of your programming code for empirical exercise as an appendix.
5. When asked to report results present the answer in a table. Nothing fancy but don't simply attach a printout of the statistical program you used. You should attach the code you used to generate the results as an appendix.

Instruction

The goal of this problem set is to setup computational environment for this course. You can choose your own scientific computation software such as R, Python, Stata, Matlab, etc. You may want to use multiple software to solve the problem set. For example, Stata would be great for econometric analysis, but Matlab, R, or other software are more convenient to do the following analysis: (1) writing down the likelihood function specific to your problem, (2) solving Bellman equation in dynamic optimization problem, and (3) doing both (1) and (2) at the same time (which is what you have to do in estimation of dynamic optimization model).

If you have no prior experience in scientific programming, I recommend you to setup R/Rstudio. R has advantages of both Stata and Matlab: it has many packages for econometric analysis and is quite flexible to do your own programming (such as writing a function and doing optimization). You can find many resources online such as "R for Data Science" (`http://r4ds.had.co.nz/`) and the website by Colin Cameron(`http://cameron.econ.ucdavis.edu/R/R.html`).

The following is the minimum list of what you should be able to do when you solve problem sets in this course.

1. Basic stuff in programming: basic algebra, vector and matrix manipulation, if-sentence, and loop.
2. Reading the dataset from .csv file. (PS2 and after)
3. Create descriptive statistics (PS2 and after)
4. Run OLS/2SLS (PS2 and after)
5. Run panel estimator, i.e., within, between, random effects estimator (PS3)
6. Estimate probit and logit model using maximum likelihood estimator (PS3 and 4)
7. Write down your own function (PS3 and after)
8. Run optimization (maximization/minimization) of the written function (PS3 and after)

Question 1 -OLS and IV-

1. Read "MROZ_mini.csv" file. I post the datafile on the class website. Report descriptive statistics of the dataset.
2. Estimate the following linear regression model by OLS;

$$\log(wage)_i = \beta_0 + \beta_1 educ_i + \epsilon_i$$

   Report estimates in a table.

3. Run IV by using $fathereduc_i$ as an instrument for $educ_i$. Report results along with OLS estimates.
4. Briefly compare estimation results of OLS and IV.

Question 2 –Vector and Matrix–

1. You. Let $y_i = \log(wage)_i$ and $\mathbf{x}_i = [1, educ_i]$. And

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, X = \begin{pmatrix} 1 & educ_1 \\ \vdots & \vdots \\ 1 & educ_N \end{pmatrix}$$

   and the OLS estimator is given by

$$\hat{\beta} = (X'X)^{-1}X'Y$$

Using programming language, calculate this matrix, and verify that you get the same number as in the one in question 1.

2. (Optional) do the same thing for IV estimate.

Question 3 - Loop-

Let $P = \{p_{i,j}\}$ be 2×2 transition matrix of a Markov process. $p_{i,j}$ is the probability that the system moves from state i to j. Consider two states $s \in \{good, bad\}$. Denote π^t be the probability mass function of the system over two states in time t. Then,

$$\pi_j^{t+1} = p_{good,j}\pi_{good}^t + p_{bad,j}\pi_{bad}^t \; j \in \{good, bad\}.$$

Thus, we can write the following matrix notation

$$\begin{pmatrix} \pi_{good}^{t+1} \\ \pi_{bad}^{t+1} \end{pmatrix} = \begin{pmatrix} p_{good,good} & p_{bad,good} \\ p_{good,bad} & p_{bad,bad} \end{pmatrix} \begin{pmatrix} \pi_{good}^t \\ \pi_{bad}^t \end{pmatrix}$$

OR,

$$\pi_{t+1} = P\pi_t$$

Consider

$$P = \begin{pmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{pmatrix}$$

Stationary distribution such that

$$\pi^* = P\pi^*$$

This is given by $(0.4, 0.6)$.

Markov chain implies that, regardless of the initial state π_0, π_t eventually converges to π^*. We numerically verify this by taking the four steps.

1. Pick arbitrary value of π_0. Note that the sum of two elements in π_0 should be 1 because this is probability mass function.

2. Using loop, show how π_t converges to π^* as t is getting larger and larger.

3. Plot $\pi_t$and t .

4. Try several different value of π_0 to see π_t converges to π^* .

Question 4 - Optimization-

Consider the following "Banana" function

$$f(x_1, x_2) = 10(x_2 - x_1)^2 + (1 - x_1)^2$$

1. Plot the value of $f(x_1, x_2)$ for the range $x_1 \in [-1, 1]$ and $x_2 \in [-1, 1]$ in the 3D plot.
2. Write down function in your programming. Numerically verify that (1) $f(1, 1) = 0$, (2) $f(2, 0) = 41$
3. Run minimization algorithm and find the point where $f(x_1, x_2)$ is minimized.