Problem Set 1

Course director: Fabio Blasutto
5329: Inequality, Household Behavior, and the Macroeconomy

Problem 1. Consider the dataset *psid1999.dta*, available on canvas in the directory — Home/ General information, data and documents/data—. You can find a description of this dataset in the file *psid1999_description.docx*, available in the same folder. Please try to execute the instructions below using the software Stata (or others if you prefer).

1.1 Change the working directory to where the data set is located on your computer, then load the dataset *psid1999.dta*.

1.2 Keep only household heads older than 24 and younger than 66 years old. Also, keep only households where the household head is married and his/her spouse is present in the family unit. Note: check the variable labels and value labels to understand which variables you should pick.

1.3 Create a variable representing the natural logarithm of the hourly earnings of the household head. Then compute the average log hourly earnings of the household head by age and by education. Consider three education categories: (i) less than high school, (ii) high school, (iii) college.

1.4 Create a scatter plot displaying average log hourly earnings of the household head over age and by education. Age should be displayed on the horizontal axis, while average hourly earnings should be on the vertical axis.

Please hand in your **.do* file with your code (or something equivalent if you use other software.)

Problem 2. Please solve the following problems using Python or a similar programming language.

2.1 Using the Python library NumPy, create a program to add a vector to each column of a given matrix. Choose the matrix and vector you prefer, your only constraint is that the number of rows of the matrix should be equal to the length of the vector.

2.2 Create a function that performs the same computations required by exercise 2.1. The function should throw an error if the number of rows of the matrix is not equal to the length of the vector. Then, call the function you created using one matrix and one vector of your choice.

2.3 The computations you performed in 2.1 and 2.2 should give the same result if the inputs are the same. Create a function that checks if the results are the same. The arguments of the function should be two matrices.

2.4 Create a class with:

- a *matrix* attribute that stores the matrix you want to modify (data).
- a *modify* method that adds a vector to each column of the matrix attribute (same function as in 2.2).

Then, use this class to create an instance matrix. Then, modify it using the *modify* method. Use the function you created in 2.3 to check whether you obtain the same modified matrix as in 2.2 and 2.1.

Please hand in your **.py* file with your code (or something equivalent if you use other software.)