# SPB4A7 Seminar 2

## MT Week 2

## Overview:

We will use a dataset from a randomised controlled trial/natural field experiment conducted by Marianne Bertrand and Sendhil Mullainathan, who sent 4,870 fictitious resumes out to employers in response to job adverts in Boston and Chicago in 2001. The resumes differ in various attributes including the names of the applicants, where some of the names are distinctly white sounding and some distinctly black sounding (in the US context).

During the seminar, we will first learn how to set up and then go through the different components of Stata (including the interface). Importantly, we will learn how to make a record of your analysis through do-files and log-files. After this, we will go through some key commands that will enable to explore descriptive statistics.

## Setting up:

The following steps are the standard setting up protocol for every seminar each week.

1. Log into your LSE Networked computer using your LSE IT username and password.
2. Go to <https://moodle.lse.ac.uk/> and log into your PB4A7 Moodle page using your LSE IT username and password.
3. Download the Seminar1.dta file and save in on your H: space. Note: it may be useful to have a PB4A7 folder with weekly sub-folders for each computer lab session.
4. To start Stata, click the start button and then type “Stata” into the finder bar; double-click on the Stata 17 icon when the options appear. Alternatively, click on the Start button at the bottom left corner, and select All Programs, then Specialist and teaching software, Statistics, Stata15.

## See your working directory:

1. Type pwd to see your working directory
2. To change your working directory to avoid typing the whole path when calling or saving files, type:

cd c:\mydata

c:\mydata

cd c:\mydata

1. Use quotes if the new directory has blank spaces, for example cd “h:\stata and data”

## Components of Your Stata Program

1. **.do Files:** These files are your programs. Do files allow you to save the commands your use and run them whenever you like. For each computer lab session/seminar, you will have to write .do files.
   * Type in *doedit* in your command window.
   * Or click the do-file editor icon to open an empty .do file. Save it in your H: space folder for PB4A7 week 1 (in the same folder with the Seminar1.dta file) by titling it (e.g. “Seminar1do.do”).
   * Or by clicking on “Do…” in the “File” menu within Stata and clicking on the .do file.
   * Or you can write your programs in WordPad, textpad, or the Stata program editor as well, buy just be sure to save your files with the .do extension.
   * If you have already .do file written, you can run it by typing the following in the Stata command window: *do filename.do, clear*
   * You can also type and run commands once at a time in the Stata command window. This is a good way to try commands and see what they do.
2. **.dta Files:** This is the file containing your Stata format dataset. Our data set’s name is “Seminar1.dta”.
   * After setting the working directory and folder path, in your .do file, you will tell Stata to use Seminar1.dta by typing: *use Seminar1, clear*
   * *clear* tells Stata to erase the previous dataset. Stata also reads non-.dta formats with *infile* or *insheet* commands.
   * You can also open your data file from within Stata, by going to *File🡪 Open 🡪 Seminar1.data.*
   * Note: that directly double-clicking Seminar1.dta will open the .dta file in an older and separate programme version of Stata, so it is recommended that you open it from within the Stata program.
3. **.log Files:** This is your output file. If you are running a lot of commands at once, it is also a good way to review your program to see what each command did. You can open your .log file with WordPad or Textpad.
   * At the beginning of your .do file, type: *log using filename.log*
   * This will create the file “filename.log” in your working directory.
   * To add more output to an existing log file, add the option append, type: *log using mylog.log, append*
   * To replace contents of your existing log file, add the option append, type: *log using mylog.log, replace (replace* tells Stata to write over the previous log while when you run the program anew – so you will lose your previous log file data).
   * At the end of the .do file, type *log close*

## Using do-files to organise your work:

While you can use Stata interactively, most work is done in .do files to aid in editing mistakes and saving your work for future reference. The lines that begin with an “\*” or “//” are ignored by Stata; these are useful for making comments to yourself. When lines do not start with “8” or “//” then Stata treats them as commands.

The typical structure of a .do file is as follows:

\*\* Filename\*\*

capture log close

set more off

set mem 700m

cd “H:”

log using filename.log, replace

use filename, clear

…

…

…

log close

* Everything between the log using line and the log close line will be saved in the .log file
* *capture* in front of a command line tells Stata to ignore the line if there is an error associated with it
* *replace* in the log using line tells Stata to overwrite the log file if it exists… be sure this is what you want
* *set mem XXX* allocates memory to Stata. If you get an error message “insufficient memory,” then more memory needs to be allocated until your program works, or your computer breaks
* *set maxvar XXX* allocates number of variables to Stata
* By default, Stata assumes that each line in the .do file is a separate command; alternatively, you can tell Stata to interpret a “;” as the end of each command line… this is done by typing:

#delimit ; \*\*turns on the ; option

#delimit cr \*\*turns the ; option off

* To save a data file from Stata go to file – save as or just type: *save, replace*
* If the dataset is new or just imported from other format go to file –> save as or just type: *save filename newfilename*

## Findings your way around your dataset

* Looking at your dataset: You can use either the Data Browser or Data Editor (only the latter allows you change values of variables). When scanning data, use the former to mitigate any mistakes.
  + An important step is to make sure variables are in their expected format. Stata has a colour-coded system for each type. Black is for numbers; red is for text or string and blue is for labelled variables.
* Quick way of finding variables: You can use the command *lookfor* to find variables in a dataset, for example you want to see which variables refer to education, type:
  + lookfor education
* The codebookcommand is a great tool for getting a quick overview of the variables in the data file. It produces a kind of electronic codebook from the data file. Have a look at what it produces.
  + It always recommended to use the codebook that comes with the dataset to have a better idea of where things are.
* To get a general description of the dataset and the format for each variable type:
  + describe
  + It tells us the number of observations in the file, the number of variables, the names of the variables, and more.
* Provide a quick overview of data file:
  + inspect
* List out the variables education, gender, black and years of experience:
  + list education female yearsexp

## Labelling variables in your dataset

* Assign a label to the data file currently in memory.
  + label data "BM2004 data"
* Assign a label to the variable female.
  + label variable female "gender"
* Create the value label genderlab and assign it to the variable foreign.
  + label define genderlab 0 "males" 1 "females"
  + label values gender genderlab

1. The data set contains a variable, yearsexp, with the number of years of work experience on the resume. Graph the historgram to see how the variable yearsexp is distributed using: histogram yearsexp.
   * For the y-axis, the “density” option scales the height of the bars so that the sum of their areas equals 1. You can use the “percent” option to scale the height of the bars so that the sum of their heights equals 100 instead.
2. To get to know some descriptive or summary statistics using: summarize yearsexp, detail.
3. Make a table of experience using the following command: tabulate yearsexp
   * Use the table command: table yearsexp. What is the difference?
4. Now look at the dummy variables (0-1 values) for female. Tabulate these variables by using the command: tab female
   * What is the share of men and women in the sample?
5. Now summarise years of experience by female: tabulate female, sum(yearsexp).
   * Do males or females have more years of experience in the sample?
   * Does the standard deviation for the variable for years of experience look similar?
6. Produce summary statistics of education, female, yearsexp, black: summarize education female yearsexp black. You can produce detailed summary statistics of education, female, yearsexp: summarize education female yearsexp black, detail.
   * What statistics describe the central tendency?
   * What statistics describe the spread of the data?
7. Produce summary statistics for yearsexp separately for males and females by using the following commands:

sort female

by female: summarize(yearsexp)

* + Does yearsexp vary by racial group (variable “black”)?

1. Graph the distribution of frequencies of years of work experience using: hist yearsexp, normal
   * Does the distribution look normal?
2. Graph the distribution of frequencies of males and females using: hist female
   * Does the graph change with hist female, discrete? Which one seems more accurate?
3. Graph the histogram with the percentage of black using: hist black, percent discrete
   * What does the distribution tell us?