Video lecture

https://youtu.be/vN3PY8KpRaM

What?

One of the most important aspects of finance is data. How we access data and how we process them becomes vital. Spreadsheet softwares (ex. MS-Excel, Apache OpenOffice and Google Docs) make data access and data processing possible. We also prepare nice looking tables and conduct simple statistical tasks etc. In fact, most of these spreadsheet applications provide a programming language in the background (ex. VBA). We must learn how to use at least one of these spreadsheet applications as part of our finance curriculum.

Advanced data management and processing however have a completely different meaning for finance. Imagine conducting a simple annual return study for all of the stocks included in the S&P-500 index for the past 10 years. For this task we would need to download daily prices for the 500 stocks for the past 10 years. Then, we would need to calculate annual returns for each of the 500 stocks and for each of the past 10 years. Doing this with a spreadsheet application becomes incredibly time consuming. We need to automate this process. We need to code.

Econometric softwares such as Stata allow this task to be completed with a fairly simple programming code. They are fast and quite reliable. More importantly, these softwares are used by the financial industry.

Why Stata?

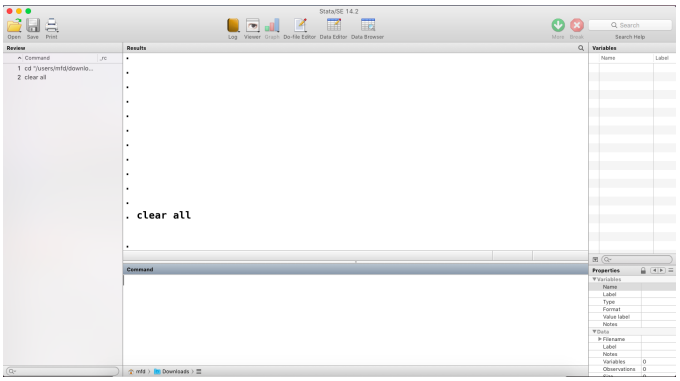
Stata is an econometric software. Because it is based on time-series data, it is quite fitting for financial econometric analysis. You can visit Stata's website for "Why Stata?" question as well https://www.stata.com/why-use-stata/.

For our purposes, Stata:

- is primarily time-series data oriented
- is incredibly fast and accurate
- has fairly simple coding language
- has extensive help available
- has operating system like environment which allows accessing online data
- allows user written commands
- has an alternative Matrix language
- has student pricing and allows renting the software

What does it look like?

Stata screen looks like the following:



Stata has many different color schemes and your Stata session may have different colors.

Stata is a command line software. It means, we type our commands to Stata and Stata does what we tell it to do. We can also give our commands to Stata in a text file but we will discuss this option a little later. For now, notice the area on Stata screen that says "Command". This is where we write our commands to Stata. The area right above (where we have the dots and the previous command "clear all") is the area of communication with Stata. It is titled "Results". Stata will repeat our commands here and it will provide results based on our commands in this area.

Left side of the screen is a log file of past commands to Stata. Notice that I had two commands. The first line is to define the working directory. The second line is to clear everything from memory. It is possible to click on these lines which will simply copy-paste them into the "command" area.

Upper right side of the screen titled "Variables" provides the list of variables in Stata currently available to us. We can load variables to Stata or generate new ones.

Lower right side of the screen titled "Properties" provides the properties of variables selected by us from the list of variables. There are several different types of variables. Each variable can have labels, definition etc. Size of each variables is also available through this area.

Hello world!

As it is custom in learning any programming language, let's say hello to the world.

```
display "Hello world!"
```

The command *display* is a Stata command that simply displays what follows. In this case a simple sentence. Notice the command line and notice the results.

User written commands

One of the most powerful features of Stata is the user-written commands. It is possible to install Stata commands provided via users of Stata. Stata has an academic journal: https://www.stata-journal.com. Many academics will share their unique programs with other users through this journal.

Let's install one of my commands: *fetchyahooquotes*. This command downloads daily prices for financial securities from Yahoo! Finance.

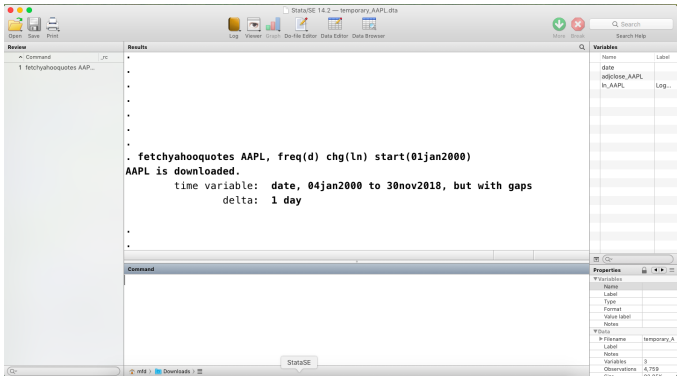
```
net install http://researchata.com/stata/203/fetchyahooquotes.pkg, force
```

If you need help with the command, just ask Stata for help.

First exercise

Now that we installed the *fetchyahooquotes*, let's download the daily prices for AAPL stock.

```
fetchyahooquotes AAPL, freq(d) chg(ln) start(01jan2000)
```

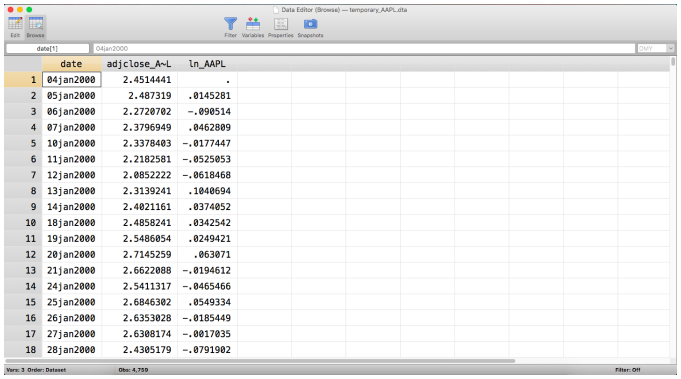


Notice that we now have three variables: *date*, *adjclose_AAPL* and *ln_AAPL*.

Data browser

Let's take a look at our data.

```
browse
```



	date	adjclose_A~L	ln_AAPL
1	04jan2000	2.4514441	.
2	05jan2000	2.487319	.0145281
3	06jan2000	2.2720702	-.090514
4	07jan2000	2.3796949	.0462809
5	10jan2000	2.3378403	-.0177447
6	11jan2000	2.2182581	-.0525053
7	12jan2000	2.0852222	-.0618468
8	13jan2000	2.3139241	.1040694
9	14jan2000	2.4021161	.0374052
10	18jan2000	2.4858241	.0342542
11	19jan2000	2.5486054	.0249421
12	20jan2000	2.7145259	.063071
13	21jan2000	2.6622088	-.0194612
14	24jan2000	2.5411317	-.0465466
15	25jan2000	2.6846302	.0549334
16	26jan2000	2.6353028	-.0185449
17	27jan2000	2.6308174	-.0017035
18	28jan2000	2.4305179	-.0791902

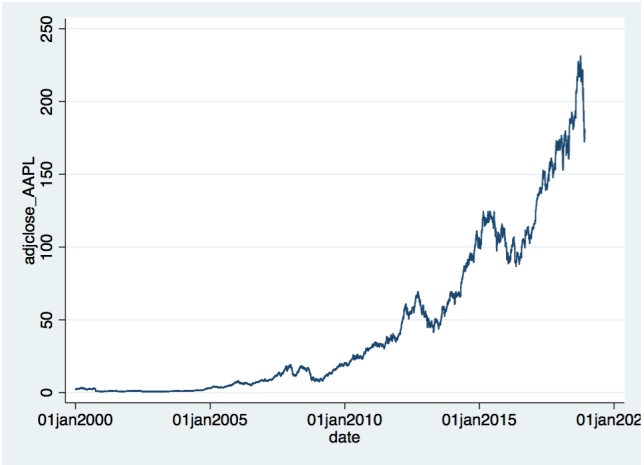
Notice that a new window will open. This is the data browser. This looks very similar to our usual spreadsheet applications. The main difference is while we can edit the data, we cannot write formulas into these cells with Stata.

Variables are in the columns: *date*, *adjclose_AAPL* and *ln_AAPL* Each row is an observation.

Simple chart

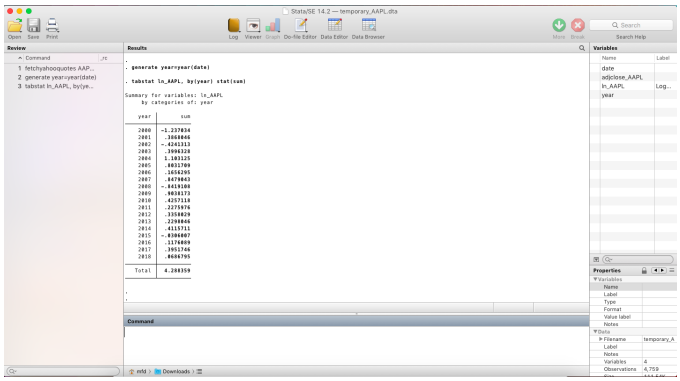
Lets draw a daily price chart for AAPL stock using our downloaded data.

```
twoway (line adjclose_AAPL date)
```



Simple financial analysis

```
generate year=year(date)
tabstat ln_AAPL, by(year) stat(sum)
```



Notice that we generate a new variable: *year*. Then, we calculated the total annual return for AAPL stock for each year since year 2000.
Conducting with analysis for all 500 stocks included in the S&P-500 index would require a few more lines of work.