Lec 8: Intro to STATA and R

1 What is STATA, and what is R?

		STATA	R
Commonalities		Tools used to perform statistical operations on computers. Available across platforms. Actively used by researchers across many disciplines.	
Differences	Nature of tool	Software	Programming language
	License	Proprietary	Open source
	Price	Costs money; UW has free access	Free
	Better for ...	Already cleaned data Little data manipulation	Data still needs to be cleaned Complex data manipulation
	Speed	Faster	OK

- You can pretty much achieve the same outcome by using either STATA or R, though the level of difficulty and the amount of time it would take differ between the two.
- I personally recommend:
  - Using R to do the data cleaning (i.e., getting data into the shape that you want, such as converting numbers coded as strings back to numerical formats, renaming columns / variable names, removing observations following certain criteria due to poor data collection, etc.)
  - Using STATA to obtain the statistics of interest.
- I don't expect you to become an expert in STATA and/or R by the end of today's lecture. If anything, our lecture today will only scratch the surface of what STATA and R can accomplish.
  - Some classes that you can take at UW:
    - SSCC training class: https://sscc.wisc.edu/sscc_jsp/training/
  - Online classes that you have access to:
    - SSCC online curriculum: https://www.sscc.wisc.edu/statistics/training/
    - LinkedIn Learning: https://lnkd.in/eFrnwki
      - Introduction to STATA 15
      - Learning R
      - R for Data Science: Lunch Break Lessons
  - What if you need help?
    - I will show you how to use the built-in help function in both STATA and R in today's lecture. Often times, this should be the first step to try when troubleshooting your code.
    - For STATA only, the company that made STATA offers instruction manual that contains some code examples. I will show you how to access such manual from the help function.
    - Google and Stack Overflow is always your friend.

2 Get ready

- Before proceeding with the rest of this worksheet (which will be us practicing using STATA and R together), you need to have STATA and R installed on your computer.
- Given that R itself is only a programming language, and we need some software to help us more easily run and troubleshoot our R code, we need to install an additional software called **R Studio**, which is an IDE (Integrated Development Environment).
  - If you need any help installing STATA, here's a quick installation guide:
    `https://go.wisc.edu/8crw4k`
  - If you need any help installing R and R Studio, here's a quick video:
    `https://youtu.be/3s57Swwoj-A`
- We will be working with a set of data that records college student's GPA. You'll need to first download this data onto somewhere on your computer. Remember where you saved the data.

  The data is available for download at the following link:

  `https://go.wisc.edu/q018t8`

  The data variable codebook is available here:

  `https://go.wisc.edu/tsy7p1`

3 Practice using STATA

1. Launch STATA. Let's go through the user interface together.
2. Create a new Do file, and use this Do file to run all of your commands below.
3. Before running any commands, let's set up the working directory to tell STATA which folder on your laptop should STATA read data and save graphs or results to.

   The easiest way to do so is to go to the menu bar, and select

   ```
   File > Change working directory...
   ```

   In the system file explorer window that popped up, navigate to the folder where you saved the GPA data, and then hit the `Choose` button.
4. We are now ready to import the data into STATA. Since the data file is named as `GPA.csv` , where the `csv` extension indicates that the data is formatted as Comma-Separated Values, we need to import the data into STATA environment as a delimited file.

   Since we don't really know what commands we should use to import such file, let's start utilizing the `help` function. Try the following command first:

   ```
   help import
   ```

   Upon examining the help document related to import commands, see if you can arrive at the final command that we need to import this file, which should be

   ```
   import delimited "GPA.csv", clear
   ```
5. Let's quickly browse the data set. Try the following two commands:

```
browse
describe
```

6. Find the mean, standard deviation, and 75th percentile of high school GPA:

   ```
   summarize hsgpa, detail
   ```

7. Generate a histogram of high school GPA:

   ```
   histogram hsgpa
   ```

8. Calculate the correlation between high school GPA and ACT score:

   ```
   correlate hsgpa act
   ```

   Interpret the correlation.

9. Create a scatterplot of high school GPA and ACT score, with ACT score on the vertical axis:

   ```
   scatter act hsgpa
   ```

10. Perform t-test to test whether the population mean of high school GPA is equal to 3.5 using a 10% size of test:

    ```
    ttest hsgpa=3.5, level(90)
    ```

4 Practice using R (working through the same exercises)

1. Launch R Studio. Let's go through the user interface together.
2. Create a new R script, and use this script to run all of your commands below.
3. Before running any codes, let's set up the working directory to tell R which folder on your laptop should R read data and save graphs or results to.

   The easiest way to do so is to go to the menu bar, and select

   ```
   Session > Set Working Directory > Choose Directory...
   ```

   In the system file explorer window that popped up, navigate to the folder where you saved the GPA data, and then hit the `Open` button.
4. We are now ready to import the data into R. Since the data file is named as `GPA.csv` , where the `csv` extension indicates that the data is formatted as Comma-Separated Values, we need to import the data into R environment as a delimited file.

   Here is where R differs significantly from STATA. STATA always works with datasets, so there is no question that STATA expects you to import a set of data.

   However, R is a programming language, so it can work with all sorts of data structure (dataset, vector, matrix, etc.). To import a data set into R, we need to make sure that it is imported as a data structure called `data.frame` .

   It is a little tricky to look up help command directly within R at this stage, but a quick Google should direct us to the following webpage:

   https://www.statmethods.net/input/importingdata.html

   From here, see if you can arrive at the final line of code to use, which should be something like

   ```
   df = read.table("GPA.csv", header=TRUE, sep=",")
   ```
5. Let's quickly browse the data set. Before doing so, I want to highlight another difference between STATA and R. STATA comes with a lot of built-in functions, since STATA is meant as a software packages.

   R, on the other hand, is an open-source language, which means that, though R comes with a lot of built-in capabilities, its power is limited until calling third-party packages that enhance R's capability.

   To help us browse the data more efficiently, I'd like for us to install such a third-party package called `psych` . To install and load this package into R's environment, try

   ```
   install.packages("psych")
   library("psych")
   ```

   Let's now try the following two commands:

   ```
   View(df)
   describe(df)
   ```
6. Find the mean, standard deviation, and 75th percentile of high school GPA:

   ```
   mean(df$hsGPA)
   sd(df$hsGPA)
   quantile(df$hsGPA, probs=.75)
   ```

7. Generate a histogram of high school GPA:

   ```
   hist(df$hsGPA, xlab="High School GPA", main="Histogram of High School GPA")
   ```

8. Calculate the correlation between high school GPA and ACT score:

   ```
   cor(df$hsGPA, df$ACT)
   ```

   Interpret the correlation.

9. Create a scatterplot of high school GPA and ACT score, with ACT score on the vertical axis:

   ```
   plot(df$hsGPA, df$ACT, xlab="High School GPA", ylab="ACT Score", main="Scatterplot")
   ```

10. Perform t-test to test whether the population mean of high school GPA is equal to 3.5 using a 10% size of test:

    ```
    t.test(df$hsGPA, mu=3.5, conf.level=.90)
    ```