Getting Started with R (Fall 2020 PLSC 498)

Angel Villegas-Cruz
Pennsylvania State University

In this brief handout, I will explain some of the basics of getting started with R. R is a free software environment for statistical computing and graphics. Programming can seem like an intimidating and impenetrable subject. But do not worry! It is actually a very intuitive and easy process.

How to install R and RStudio?

For Windows Users:

1. Open an internet browser and go to https://www.r-project.org/.
2. Click the "download R" link in the middle of the page under "Getting Started."
3. Select a CRAN location (a server closer to your physical location is preferred) and click the corresponding link.
4. Click on the "Download R for Windows" link at the top of the page.
5. Click on the "install R for the first time" link at the top of the page.
6. Click "Download R for Windows" and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.
7. Now that R is installed, you need to download and install RStudio. R-Studio is an environment for R.
8. Go to https://rstudio.com/and click "Download" at the top right corner of the page.
9. Click on "Download RStudio Desktop" (Open Source License).
10. Click on the version recommended for your system (Windows), and save the executable file. Run the .exe file and follow the installation instructions.

For Mac Users:

1. Open an internet browser and go to https://www.r-project.org/.
2. Click the "download R" link in the middle of the page under "Getting Started."
3. Select a CRAN location (a server closer to your physical location is preferred) and click the corresponding link.
4. Click on the "Download R for (Mac) OS X" link at the top of the page.
5. Click on the file containing the latest version of R ("R-3.6.2.pkg") under "Latest Release."
6. Save the .pkg file, double-click it to open, and follow the installation instructions.
7. Now that R is installed, you need to download and install RStudio. R-Studio is an environment for R.
8. Go to https://rstudio.com/and click "Download" at the top right corner of the page.

9. Click on "Download RStudio Desktop" (Open Source License).
10. Click on the version recommended for your system (macOS), double-click it to open, and then drag and drop it to your applications folder.

The basics

```
#R is case sensitive
#Comments are code that begin with "#" and aren't run when you run an .R file.
#Comment your code frequently as you work so you can easily return to your document
and remember what you did.
#Define Variables:
x <- 7 #this means you "assign" 7 to an object called x
x <- 7
y <- 5
x+y #addition
[1] 12
x-y #subtraction
x*y #multiplication
x/y #division
x^y #exponent
sqrt() #square root
abs() #absolute value
log() #log
```

How to read data into R?

Step one, preparing your data set:

- Before you start thinking about how to load your files and spreadsheets into R, you need to first make sure that your data is well prepared to be imported.
- The first row of the spreadsheet is usually the header. For instance, if you have a column of Asian states by GDP per capita, the first row should say "gdp" or "gdpasia ".
- Avoid names, values or fields with blank spaces. If you want to include multiple words, insert a dot (".") or underscore ("_").
- Make sure your missing values are empty or indicated with "NA".
- Avoid using names that contain symbols such as "!" or "$".

Step two, downloading and saving your data set:

- Let us try with a real data set. Click **here** and download "DataAnalysis". DataAnalysis is an Excel document ("xlsx" file). Excel offers many options to save your file besides the default extension "xlsx". Below I will illustrate how to save and upload into R the following formats: "xlsx", ".csv", ".txt", and ".dta":

a- Using "xlsx":

- After you download "**DataAnalysis**", save it in a folder named "PLSC 498" in your computer's desktop. Once you have your data set saved, you need to set your working directory in R. The directory is where R goes to look for files and save your progress.

```
getwd() #get working directory
setwd() #set working directory
#try to set "PLSC 498" folder as your working directory by writing the code below
setwd("/Users/Desktop/PLSC 498") #set our "PLSC 498" folder as a working directory
#the information between the quotes ("") could be different in your computer
#the information between the quotes should be the location of the "PLSC 498" folder
```

- If you correctly set up "PLSC 498" as your working directory in R, you should see the folder with all its files at the bottom left corner of the screen under "Files".
- To upload the "DataAnalysis" data set into R write the code below:

```
library(readxl)
DataAnalysis <- read_excel("DataAnalysis.xlsx") #importing "xlsx" file
View(DataAnalysis) #viewing the data set
```

- If you correctly uploaded "DataAnalysis" into R, you should see the data set at the top left corner of the screen under "Environment".

b- Using ".csv":

- After you download "DataAnalysis", save it in a folder named "PLSC 498" in your computer's desktop. You can convert "DataAnalysis" from a "xlsx" to a ".csv" file. To do this, click on "Save As" and select the extension "CSV", which is listed as one of the "Save as Type" options. Once your new ".csv" file data set is saved, you need to set your working directory in R. The directory is where R goes to look for files and save your progress.

```
getwd() #get working directory
setwd() #set working directory
#try to set "PLSC 498" folder as your working directory by writing the code below
setwd("/Users/Desktop/PLSC 498") #set our "PLSC 498" folder as a working directory
#the information between the quotes ("") could be different in your computer
#the information between the quotes should be the location of the "PLSC 498" folder
```

- If you correctly set up "PLSC 498" as your working directory in R, you should see the folder with all its files at the bottom left corner of the screen under "Files".
- To upload the "DataAnalysis" data set into R write the code below:

```
library(readr)
DataAnalysis <- read_csv("DataAnalysis.csv")#importing "csv" file
View(DataAnalysis) #viewing the data set
```

- If you correctly uploaded "DataAnalysis" into R, you should see the data set at the top left corner of the screen under "Environment".

c- Using ".txt":

- After you download "DataAnalysis", save it in a folder named "PLSC 498" in your computer's desktop. You can convert "DataAnalysis" from a "xlsx" to a ".txt" file. To do this, click on "Save As" and select the extension "Text", which is listed as one of the "Save as Type" options. Once your new ".txt" file data set is saved, you need to set your

working directory in R. The directory is where R goes to look for files and save your progress.

```
getwd() #get working directory
setwd() #set working directory
#try to set "PLSC 498" folder as your working directory by writing the code below
setwd("/Users/Desktop/PLSC 498") #set our "PLSC 498" folder as a working directory
#the information between the quotes ("") could be different in your computer
#the information between the quotes should be the location of the "PLSC 498" folder
```

- If you correctly set up "PLSC 498" as your working directory in R, you should see the folder with all its files at the bottom left corner of the screen under "Files".
- To upload the "DataAnalysis" data set into R write the code below:

```
DataAnalysis <- read.delim("DataAnalysis.txt") # importing "txt" file
View(DataAnalysis) #viewing the data set
```

- If you correctly uploaded "DataAnalysis" into R, you should see the data set at the top left corner of the screen under "Environment".

d- Using ".dta":

- The ".dta" format is native to Stata. One way to read ".dta" files into R is the "readstata13" package. Below I will show you how to install this package into R and read Stata data files into R.

```
#First, install the package with the code below
install.packages("readstata13")
#Second, run the code below to upload the file into R
library(readstata13)
dta <- read.dta13("DataAnalysis.dta") # importing "dta" file
```

- If you correctly uploaded the data set into R, you should see it at the top left corner of the screen under "Environment".

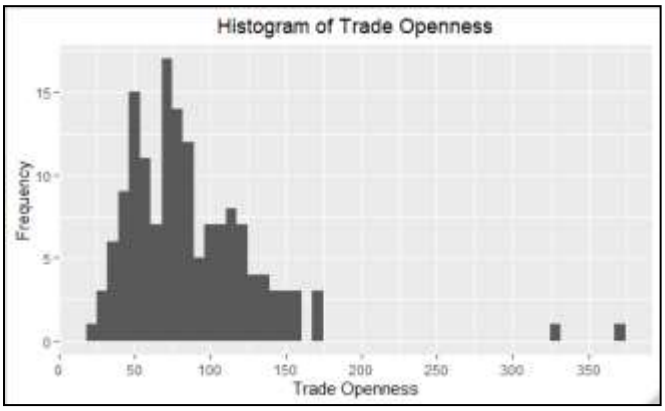
Some summary statistics

I will show you some summary statistics and basic graphs using the previous data set "**DataAnalysis**".

```
#As a reminder, I will start by opening my data set in R.
DataAnalysis <- read_excel("DataAnalysis.xlsx") #I am using a .xlsx file.
#Define Variables: You usually find the definition of variables in a dataset codebook.
#DataAnalysis has six variables: (1) country-name, (2) year, (3) polity score, (4)oecd
dummy, (5) trade openness (imports + exports as % of GDP), (6) log trade openness.
trade.openness <- DataAnalysis$openness #openness is our fifth column/variable (trade
openness)

summary(trade.openness) #the summary() function gives you summary statistics for just
one of the variables
 Min. 1st Qu.  Median    Mean  3rd Qu.   Max.
22.77   54.93   76.31   86.77  109.24  372.10
```

Let us now make some basic graphs using "ggplot2". The data visualization package "ggplot2" contains a number of functions to produce professional plots.

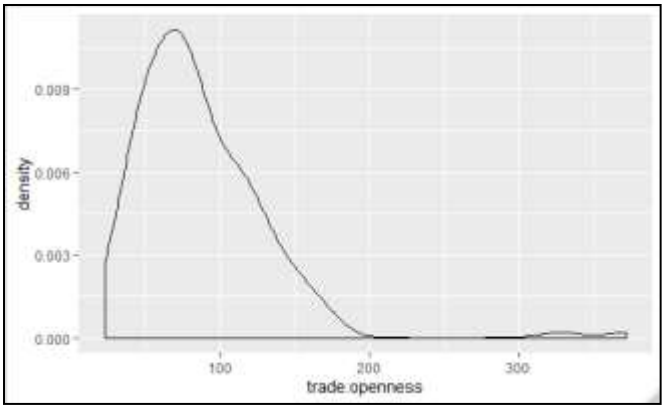
```
#First, you need to install the ggplot2 package.
install.packages("ggplot2") #Run this line of code only once
library(ggplot2) #to load ggplot2

#Histogram
ggplot(DataAnalysis, aes(x=trade.openness)) + #a simple histogram of the variable
"trade.openness"
  geom_histogram()

#Adding additional things to the histogram
ggplot(DataAnalysis, aes(x=trade.openness)) +
  geom_histogram(bins=50) + #changing the bin width
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_continuous(breaks=seq(0,400, by=50)) +
  labs(title="Histogram of Trade Openness", #add a main title
     x="Trade Openness", #add x-axis title
     y="Frequency") #add y-axis title
```




```
#How to export a plot?
#To save a ggplot is very simple. Below I will save plot1.
plot1 <- ggplot(DataAnalysis, aes(x=trade.openness)) +
  geom_histogram(bins=50) + #changing the bin width
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_x_continuous(breaks=seq(0,400, by=50)) +
  labs(title="Histogram of Trade Openness", #add a main title
       x="Trade Openness", #add x-axis title
       y="Frequency") #add y-axis title

#Run the code below to save plot1 as a pdf in your working directory.
ggsave("BoxplotExample.pdf", plot= plot1, width=5, height=5, units="in")
```

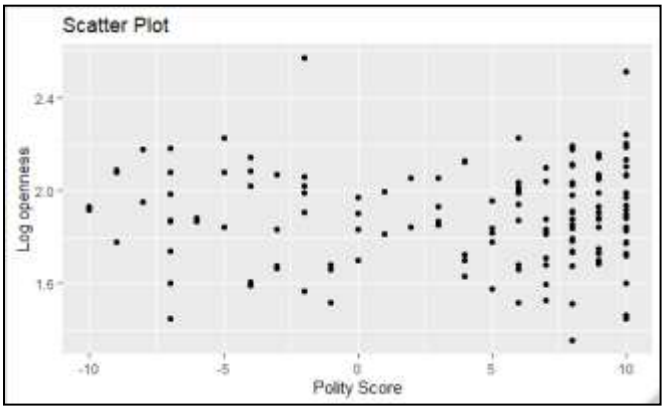
Now, I will show you how to do a density plot. A density plot shows you the distribution of data over a continuous interval or time period.

```
#Density Plots
ggplot(DataAnalysis, aes(x=trade.openness)) +
  geom_density()#function for making a basic density plot
```



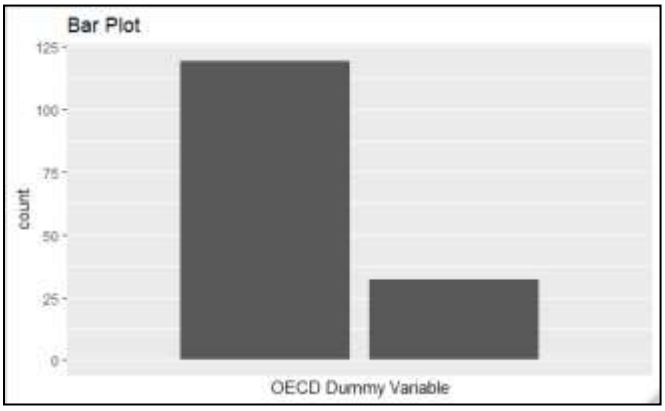
Other useful summary statistics functions are:

```
#Bivariate Correlation
cor() #This function computes the correlation coefficient.
#A bivariate correlation test needs two variables. Below I will define my independent
variable and dependent variable.
polity.score <- DataAnalysis$polity2 #The Polity Score will be my independent variable.
log.openness <- DataAnalysis$logopenness #Log Openness will be my dependent variable.
cor(polity.score, log.openness) #example using our previous variables
[1] -0.01440345 #This is the correlation coefficient.

#Scatter Plot
#A scatter plot needs two variables (x and y). Below I will define my x (independent
variable) and y (dependent variable).
polity.score <- DataAnalysis$polity2 #The Polity Score will be my independent variable.
log.openness <- DataAnalysis$logopenness #Log Openness will be my dependent variable.
ggplot(DataAnalysis, aes(x=polity.score, y=log.openness)) +
  geom_point() +
  labs(title="Scatter Plot", #add a main title
       x="Polity Score", #add x-axis title
       y="Log openness") #add y-axis title
```

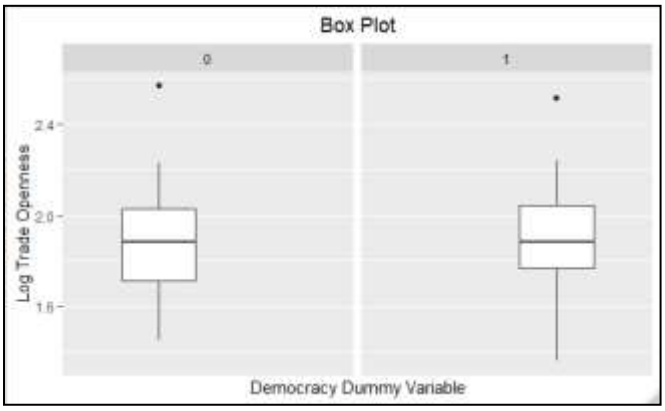



```
#Bar Plot
#Let us make a bar plot of our fourth column/variable
oecd.variable <- DataAnalysis$oecd #oecd is our fourth column/variable (oecd dummy)
#Making a bar plot of oecd.variable
ggplot(DataAnalysis, aes(x=oecd.variable)) +
  geom_bar(bins=200) +
```

```
  labs(title = "Bar Plot", x = "OECD Dummy Variable") + #add a main and x-axis title
  scale_x_discrete() +
  scale_y_continuous(limits=c(0,120))
```




```
#Box Plot
#A box plot is an alternative way to visualize the distributions of an interval-level
variable.
#Let's say I want to see whether trade openness is more significant in autocracies or
democracies.
#Defining my variables
log.openness <- DataAnalysis$logopenness
polity.score <- DataAnalysis$polity2

#Below I am re-coding polity.score with the "recode" function from the CAR package
library("car")
DataAnalysis$politybinary <- car::recode(DataAnalysis$polity2,"c(7,8,9,10)=1;else=0")
#The line of code above tells R to code Polity scores 7, 8, 9 and 10 as 1, while
coding everything else 0. I store it in a new column/variable called politybinary.

ggplot(DataAnalysis, aes(x=politybinary, y=log.openness)) + #making a box plot
  geom_boxplot() +
  theme(plot.title = element_text(hjust = 0.5)) +
  facet_wrap(~politybinary) +
  scale_x_discrete() +
  labs(x="Democracy Dummy Variable",
    y="Log Trade Openness",
       title="Box Plot")
```



```
#t-test
#A two-sample t-test is used to test the difference between two population means.
#Below a t-test of difference in means of log trade openness (imports + exports as %
of GDP) for democratic and autocratic countries.
t.test(log.openness[DataAnalysis$politybinary == 1],
log.openness[DataAnalysis$politybinary == 0])
#Is trade openness more significant in autocracies or democracies?
```

Other resources to learn more about R

There is a plethora of resources available online to help you with R. Here are some resources to get you started:

The Undergraduate Guide to R - a beginner's introduction to the R programming language.

DataCamp - for more information about importing data files from different sources.

Quick-R - an R introduction site.

R-Bloggers.com - a blog aggregator of content contributed by bloggers who write about R.

Exploratory Data Analysis in R by Pearson - a book that introduces how to use R to conduct exploratory data analysis.