R Bootcamp for Data and Policy

Justin Ho

Director of Data and Policy

Groundwork UNL

groundwork@unl.edu

September 2020

Contents

Introduction

Groundwork UNL is a student-led organization that strives to have a more data driven approach to campus policies. In this module, we are providing an introductory analysis to using the R Programming Language and Statistical software to perform some data analysis.

The data that we are using is the data collected on UNL's COVID-19 Response. Sources can be obtained in the link provided.

Source: https://github.com/groundworkunl/UNL_COVID19.

1 Data Science Project Life Cycle

Figure 1: Source: R for Data Science (Wickham and Grolemund, 2016)

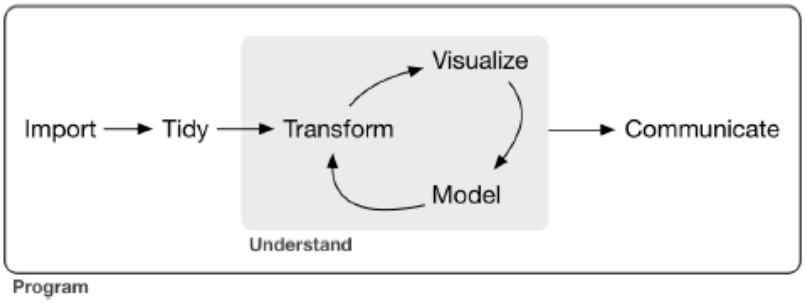

Figure 1 shows a diagram for the life cycle of a typical Data Science Project. The first step of the process is to import data, and this can either be primary or secondary data. In this situation, we are using primary data we collected from students.

The next step of the process is to tidy (aka clean, wrangle etc.) the data for it to be useable. Transformation of the data is then done, and it is visualized, modeled and the cycle repeats given new areas of interest of looking at the data arises as we go through this process.

Lastly, communication of the results takes place as an end product of the data science project. In this module, we are focusing on the tidying and cycle to understand our dataset.

2 Tidying Data

2.1 Setting Up tidyverse

In this module, we are using the tidyverse package, a package that contains the most useful packages in R for data cleaning and exploration.

If you do not have the tidyverse package, run the following code in the console.

```
install.packages("tidyverse")
```

Once you have the following package, we can proceed to import the package to be used with the following code:

```
library(tidyverse)
```

In this module, we are primarily using the tibble, dplyr and ggplot2 packages.

2.2 Cleaning Data

Cleaning data is perhaps the most exhausting process in a data science project life cycle. This is inevitable given the nature of how data is collected.

Refer to the code in TidyAndExploreCOVID19.R lines 6 to 87 for the cleaning of the data for analysis.

An important part to note for R's syntax that is not found in most other programming language is the % > % symbol (refer to lines 31 to 36). This symbol takes the object on the left and place it as the first argument of the function on the right.

In R, the two sets of code are considered equivalent.

```
sampleClass <- covid %>%  # first method
  select(class)

sampleClass <- select(covid, class) # second method
```

However, the former way of writing is arguably better aesthetically given it avoids function nesting. The lines of code from 31 to 36 can become overwhelming quickly if repeated nesting is used.

Also note that R, like most programming languages are case sensitive when matching strings of characters. This is why extensive clean up is required for different datasets to be compatible to one another.

2.3 Transforming Data

Having the actual and sample demographics, the logical next step is to compare the actual and sample demographics. This can be done by transforming some of the data using functions in dplyr.

In this module, we focus on using *summarize*(), *group_by*(), *select*() and *mutate*(). Refer to lines 31 to 35.

1. *select*()
   select takes in the dataset for the first argument, and selects the columns stated in the other arguments. Adding a − sign refers to deselect

2. *group_by*()
   this function groups the unique elements in the column and treat them as a group. This doesn't change the outlook of the dataset, but subsequent functions would be altered. For an example, if in the column for isStudent TRUE and FALSE, group_by(isStudent) would cause subsequent functions to treat TRUE to be a group and FALSE to be another

3. *summarize*()
   The summarize function reduces a data frame to a summary of just one vector or value. Many times, these summaries are calculated by grouping observations using a factor or categorical variables first.

4. *mutate*()
   The mutate function is used to create a new variable from a dataset. This generates a new column for the dataset.

In this module, the *rbind*() function is used as well. This function binds two separate dataset into one by their rows (*rbind*() short for "rowbind"). If the columns match, they would treated as the same column. It is important for the column names to be exactly the same if you intend to combine both datasets without generating new columns.

3 Visualizing Data

R has in-built functions such as *plot*() or *hist*() for generating visuals. They are in general sufficient in most elementary exploration of data.

However, a more powerful tool for data visualization can be obtained from the ggplot2 package which allows incremental changes and addition to generate more comprehensive plots as compared to the standard *plot*() function.

Due to my lack of expertise in this region, I could only share as much as I know about the package in the sample code provided. Google is an extremely great tool for any questions regarding ggplot2. Refer to code 100 to 103 for an example of generating plots.

Just for fun and to make this document pretty, this is one of the plots I generated. in this module.

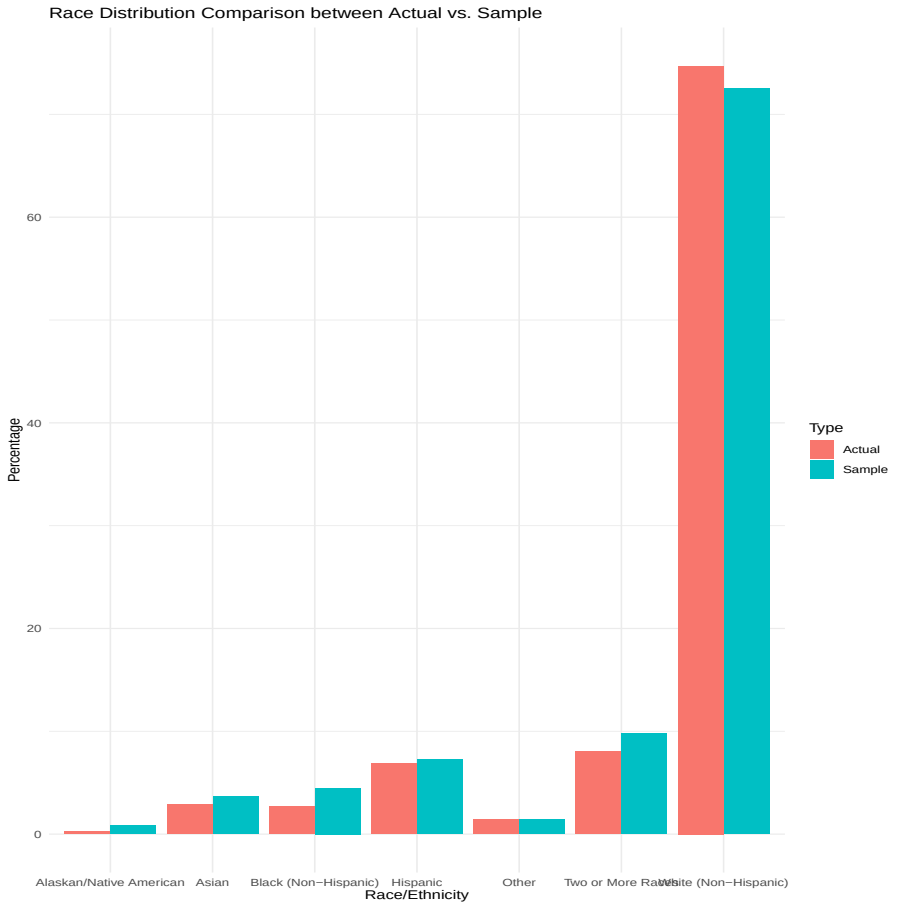

Figure 2: Caption

4 Statistical Analysis

4.1 Sample Proportion Tests

Due to the nature of the unrepresentative sampling, some questions arise as a result. For an example, given the high proportion of respondents are freshman, is it possible that due to oversampling, this causes the dataset to be unrepresentative as a whole?

Though we cannot truly answer the following question, we can obtain some insights as to whether the differences in class level could affect the dataset's generalizability to UNL's population. This can be done by using a t-test for two sample proportions.

The dataset can be subset in many ways to obtain the results we are interested in. The following code subsets the larger dataset into freshman and not a freshman.

```
covid <- covid %>%
  mutate(isFreshman = (class == "Freshman"))

covid.f <- covid[covid$isFreshman,]
covid.o <- covid[!covid$isFreshman,]
```

Since this is not a class on statistics, nor am I the most versed in statistics, we are using a two-tailed test for all of the tests.

4.1.1 Abstractions with Functions

Observe the following code:

```
t_test <- function(data_a, data_b){
  # Remove NA from vector
  data_a <- data_a[!is.na(data_a)]
  data_b <- data_b[!is.na(data_b)]

  mean.1 <- mean(data_a)
  cat("Mean of A: ", mean.1, "\n")
  mean.2 <- mean(data_b)
  cat("Mean of B: ", mean.2, "\n")
  sd.1 <- sd(data_a)
  sd.2 <- sd(data_b)
  n.1 <-length(data_a)
  n.2 <-length(data_b)

  # Now construct the test statistic
  se <-sqrt((sd.1^2)/n.1 + (sd.2^2)/n.2)
  t <-(mean.1-mean.2)/se
  cat("t-score: ", t, "\n")
  p <- 2*((1-pt(abs(t), df=n.1+n.2-2)))
  cat("P(|t| >= ", abs(t), ") = ", p ,"\n")
}
```

The following code creates a function that accepts two inputs and generate the results of the two sample proportions two-tailed t-test. This allows a more modular and reusable code that is more efficient in general.

4.1.2 Results from t-tests

Note that performing t-tests between freshman and non-freshman students yielded mostly insignificant results except for tuition. This does not prove necessarily that freshman students and non-freshman would answer similarly, but it is not very likely they would significantly.

Repeat this with gender, college and race. This allows a more thorough understanding of the results obtained and raises questions to be answered in the future.

4.2 Modelling Data

Models are overrated, and over complicated. While it is true that models can be behemoths that take years of studying to understand, simple linear models are also useful models that help us understand the world around us and they can be created relatively easily.

Focusing on Ordinary Least Squares Regression (OLS) models, we can use the *lm*() function that does just that. Refer to the code below:

```
lmod <- lm(satisfaction ~ awareness + hygiene + mask, covid)
summary(lmod)
```

The console should output the following results:

```
Call:
lm(formula = satisfaction ~ awareness + hygiene + mask, data = covid)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0638 -0.7133  0.0703  0.6785  3.2045

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.54563    0.36529   4.231 2.98e-05 ***
awareness    0.17523    0.05466   3.206 0.001471 **
hygiene      0.56707    0.05269  10.762  < 2e-16 ***
mask        -0.23867    0.06255  -3.816 0.000161 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1

Residual standard error: 0.9606 on 348 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.3033,    Adjusted R-squared:  0.2972
F-statistic: 50.49 on 3 and 348 DF,  p-value: < 2.2e-16
```

4.2.1 Gauss-Markov Theorem

Most introductory statistics classes look at the coefficients and parameters generated, the R-squared and the significance and call it quits. However, to have a more robust understanding of the linear model generated, it is good to see if the linear model satisfies the Gauss-Markov Assumptions, which guarantee the validity of ordinary least squares for estimating regression coefficients.

The following are the Gauss-Markov Assumptions:

1. Linearity: the parameters we are estimating using the OLS method must be themselves linear.
2. Random: our data must have been randomly sampled from the population.
3. Non-Collinearity: the regressors being calculated aren't perfectly correlated with each other.
4. Exogeneity (Zero Conditional Mean of Error): the regressors aren't correlated with the error term.
5. Homoscedasticity: no matter what the values of our regressors might be, the error of the variance is constant.

In practice, the assumptions are rarely met. However, being at least close to satisfying the assumptions allow you to be confident that your linear model is BLUE (Best Linear Unbiased Estimators).

Looking at the diagnostic plots, we could at least have a good idea as to how well our model fits the Gauss-Markov Assumptions.

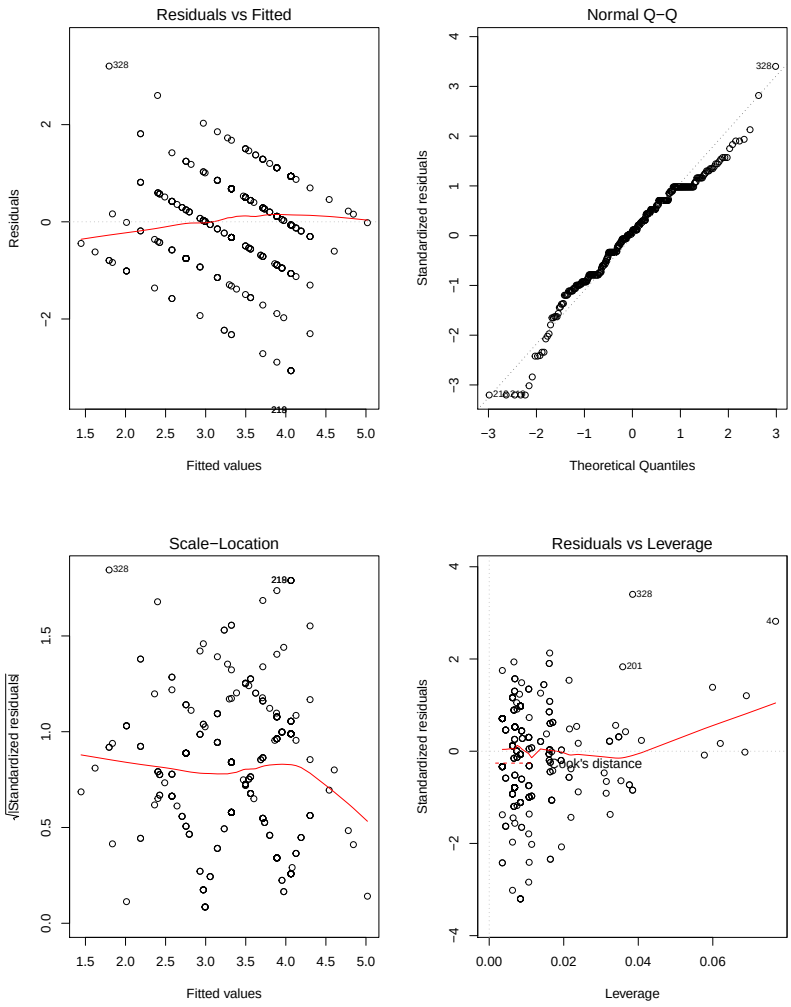

Figure 3: Diagnostic Plots for the Linear Model Above

5 Disclaimer

THIS IS NOT AN OFFICIAL COURSE OR BOOTCAMP ON R AND IT IS CONDUCTED PRO BONO. I DO NOT CLAIM THAT I AM A PROFESSIONAL IN THE FIELD OF STATISTICS, ECONOMICS, SOCIAL SCIENCES OR COMPUTER SCIENCE. THESE ARE EXPERIENCES THAT I HAVE GATHERED FROM WORKING ON RESEARCH AND I AM CONDUCTING THIS WITH THE OFFICIAL CAPACITY AS THE PRESIDENT AND DIRECTOR OF DATA AND POLICY OF THE REGISTERED STUDENT ORGANIZATION GROUNDWORK UNL.

THIS IS NOT AN OFFICIAL ASUN CORRESPONDENCE.

5.1 About Me

Excerpt from Linked-In

It seems awfully easy to make bad decisions on a daily basis, and I am guilty of plenty. That is why I am extremely curious about having a more systematic framework to think about decision making in a more rational and constructive way. This drew me into the field of economics, and with the use of large amount of data, economics drew me into computer science. I am currently working as an undergraduate researcher on economics whereby huge datasets are collected, transformed and thoroughly analyzed.

Leveraging my understanding in set theory, databases and computer science fundamentals, I was able to effectively manage the dataset that I obtained from various sources. My current research pertains largely to labor economics, specifically in human capital development. Much of human capital economics are theoretical due to the lack of substantial data to perform analysis on and with new data available, I wanted to test the empirical validity of some of these theories.

With much guidance from my professor, I was able to ask a lot of questions regarding the data we found to ensure the robustness of the model that we are building. These are framed in a lot of perspectives that sometimes yield unexpected results. I think this mental exercise of being questioning, interpreting and forming hypotheses to test the model I built allowed me to hone my critical thinking skills.

I am currently a Junior in University of Nebraska-Lincoln and keen to explore more opportunities, be it in the field of economics or computer science. I do think that having a data-driven mindset does allow us to be more productive and that is one of the reasons why I founded a student organization (Groundwork UNL) that focuses on data-driven framework to inform and affect campus policies. I am interested in a career path that allows me to utilize my skills in data science.