# Assignment 2

Derek Rouch

LLO 8200

Introduction to Data Science

September 1, 2019

## 1) Calculate the mean of the outcome.

To begin, I cleared my Global Environment and get the necessary libraries: knitr, tidyverse, plotly, and Metrics.

## Clear environment  
rm(list=ls())  
  
## Get libraries  
library(knitr)  
knitr::opts\_chunk$set(echo = TRUE)  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.2.1 ✔ purrr 0.3.2  
## ✔ tibble 2.1.3 ✔ dplyr 0.8.3  
## ✔ tidyr 0.8.3 ✔ stringr 1.4.0  
## ✔ readr 1.3.1 ✔ forcats 0.4.0

## ── Conflicts ────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(plotly)

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

library(Metrics)

Then, I loaded the pd.Rdata dataset, as well as the pd\_lab\_explain.Rdata codebook to decode the column headings.

## Load in the county-level data  
load("/Users/derekrouch/Documents/GitHub/data-science/pd.Rdata")  
  
## Load the variable descriptions  
load("/Users/derekrouch/Documents/GitHub/data-science/pd\_lab\_explain.Rdata")

With the data loaded, I was able to pipe it to the summarize function to view the unconditional average of the home ownership rate variable, homeown\_rate, which I found using the mean function.

## Unconditional average  
pd%>%summarize(mean\_homeown\_rate=mean(homeown\_rate,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_homeown\_rate  
## <dbl>  
## 1 72.7

## 2) Use your mean as a prediction: Create a new variable that consists of the mean of the outcome.

To create a new variable, I piped the data to the mutate function, and added mean\_homeown\_rate as a new column in my original pd dataset.

##Unconditional average as a predictor  
pd <- pd%>%mutate(mean\_homeown\_rate=mean(homeown\_rate,na.rm=TRUE))

## 3) Calculate a summary measure of the errors for each observation—the difference between your prediction and the outcome.

Now that every observation in my dataset had the same value for mean\_homeown\_rate, I could calculate the error for each observation—that is, how far each county’s home ownership rate is from the mean of 72.7%.

To accomplish this, I subtracted the mean\_homeown\_rate with each individual county’s homeown\_rate. I named this difference error\_homeown. Counties with positive error values have home ownership rates higher than the national average, while counties with negative error values have home ownership rates below that of the national average.

pd <- pd%>%mutate(error\_homeown=homeown\_rate-mean\_homeown\_rate)

## 4) Calculate the mean of the outcome at levels of a predictor variable.

To determine a predictor varaible, I first examined the table. However, with 58 variables (the original 56 plus my 2 new ones), it felt like I would be shooting in the dark, so I wanted a way of enhancing my guesswork.

I knew this would require venturing to the far frontiers of my *R* knowledge, but I felt that there was no corner Google and StackExchange couldn’t get me out of, so I ventured on.

*I apologize in advance if the following process is an affront to real statistical analysis.*

My first decision was to create a correlation matrix.

To do this, I first needed to make sure that the data was in numeric form. I used sapply to determine the class of each variable.

## Determine each variable's class  
sapply(pd, class)

## fips pop2013 pop2010\_base   
## "character" "integer" "integer"   
## popchange\_pc pop2010 popu5   
## "numeric" "integer" "numeric"   
## popu18 pop65p female\_pc   
## "numeric" "numeric" "numeric"   
## white\_pc black\_pc am\_ind\_pc   
## "numeric" "numeric" "numeric"   
## asian\_pc hawaii\_pi\_pc twomore\_race\_pc   
## "numeric" "numeric" "numeric"   
## hispanic\_pc white\_non\_hispanic\_pc same\_house\_pc   
## "numeric" "numeric" "numeric"   
## foreign\_born\_pc other\_eng\_home\_pc hs\_grad\_pc   
## "numeric" "numeric" "numeric"   
## coll\_grad\_pc veterans travel\_time   
## "numeric" "integer" "numeric"   
## housing\_units homeown\_rate house\_unit\_multi   
## "integer" "numeric" "numeric"   
## median\_home\_val households person\_per\_hh   
## "integer" "integer" "numeric"   
## per\_capita\_inc median\_hh\_inc persons\_below\_poverty   
## "integer" "integer" "numeric"   
## pv\_nonfarm pv\_nonfarm\_employ pv\_nonfarm\_employ\_ch   
## "integer" "integer" "numeric"   
## nonemployer\_est firms firms\_black\_own\_pc   
## "integer" "integer" "numeric"   
## firms\_amind\_own\_pc firms\_asian\_own\_pc firms\_hawaii\_pi\_own\_pc   
## "numeric" "numeric" "numeric"   
## firms\_hispanic\_own\_pc firms\_female\_own\_pc manufacture\_ship   
## "numeric" "numeric" "numeric"   
## wholesale retail retail\_percap   
## "numeric" "numeric" "integer"   
## hospitality bldg\_permits land\_area   
## "integer" "integer" "numeric"   
## pop\_per\_square county percapinc.2010   
## "numeric" "character" "numeric"   
## percapinc.2012 mean\_homeown\_rate error\_homeown   
## "numeric" "numeric" "numeric"

This showed me that the dataset’s variables fips and county were stored as text, so I used subset to remove those columns and create a new, numeric-only dataset—which I named pd\_num\_only.

## Create a subset of the data, consisting of only numeric variables  
pd\_num\_only <- subset(pd, select = -c(fips, county))

With this new dataset, I was finally able to create a correlation matrix, which I named cor\_matrix. Since I am only considered with homeown\_rate correlations, I made a subset of the matrix to include only that variable.

## Create a correlation matrix, rounded to two decimal places  
cor\_matrix <- cor(pd\_num\_only)

## Warning in cor(pd\_num\_only): the standard deviation is zero

round(cor\_matrix, 2)  
  
## Remove every column except `homeown\_rate` in the matrix  
cor\_homeown\_rate <- subset(cor\_matrix, select = c(homeown\_rate))

cor\_df <- data.frame(row=rownames(cor\_matrix)[row(cor\_matrix)[upper.tri(cor\_matrix)]],   
 col=colnames(cor\_matrix)[col(cor\_matrix)[upper.tri(cor\_matrix)]],   
 corr=cor\_matrix[upper.tri(cor\_matrix)])

## 2) R output that summarizes the variables in the college.Rdata dataset

To begin, I will clear my Global Environment and get the library for the tidyverse package, which has already been installed.

## Clear environment  
rm(list=ls())  
## Get tidyverse library  
library(tidyverse)

Then, I need to load the college.Rdata dataset.

## Load in the data  
load("~/data-science/college.Rdata")

### Finding Average Earnings by College Selectivity

To find the average earnings based on admission selectivity, I can filter out colleges according to their admission rates. I’ll define the most selective schools as those that admit less than 10% of applicants *(adm\_rate<.1)*, and the least selective schools that admit more than 30% *(adm\_rate>.3)* of their applicants.

## Average earnings for most selective colleges in dataset (admin rate of 10% or less)  
sc%>%filter(adm\_rate<.1)%>%summarize(mean\_earnings=mean(md\_earn\_wne\_p6,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_earnings  
## <dbl>  
## 1 53500

## Average earnings for least selective colleges in dataset (admin rate of 30% or more)  
sc%>%filter(adm\_rate>.3)%>%summarize(mean\_earnings=mean(md\_earn\_wne\_p6,na.rm=TRUE))

## # A tibble: 1 x 1  
## mean\_earnings  
## <dbl>  
## 1 34747.

### Comparing Average Earnings

To find the difference in average earnings between the most selective *(adm\_rate<.1)* and least selective *(adm\_rate>.3)* colleges, I can simply subtract these filtered means.

## Difference in average earnings between most selective and least selective colleges in dataset  
sc%>%filter(adm\_rate<.1)%>%summarize(mean\_earnings=mean(md\_earn\_wne\_p6,na.rm=TRUE))-sc%>%filter(adm\_rate>.3)%>%summarize(mean\_earnings=mean(md\_earn\_wne\_p6,na.rm=TRUE))

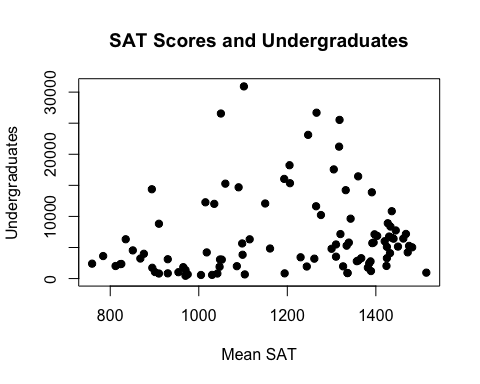
## mean\_earnings  
## 1 18752.86

This shows us that graduates from selective schools have average earnings that are $18,752.86 higher than the average earnings of graduates from less selective schools.

### SAT Scores and College Size

To determine whether colleges with very high SAT scores tend to be larger or smaller than colleges with low SAT scores, I first plotted the variables to visually inspect whether a relationship was obvious.

## Plot of SAT Scores and College Size  
plot(sc$sat\_avg, sc$ugds,main="SAT Scores and Undergraduates", xlab="Mean SAT", ylab="Undergraduates", pch=19)



Although there did not appear to be a correlation, I also ran the cor.test function to test for association between average SAT scores and the number of undergraduates enrolled.

## Pearson Correlation test  
cor.test(sc$sat\_avg, sc$ugds)

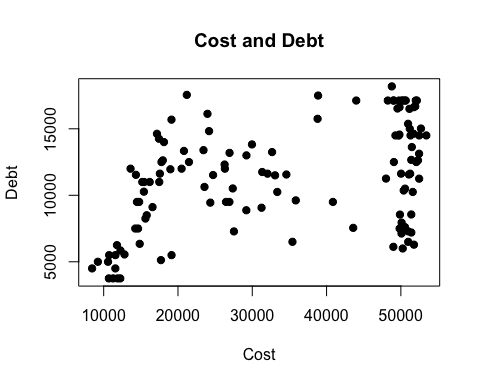
##   
## Pearson's product-moment correlation  
##   
## data: sc$sat\_avg and sc$ugds  
## t = 1.3208, df = 96, p-value = 0.1897  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.06659166 0.32344133  
## sample estimates:  
## cor   
## 0.1335943

The cor.test confirmed that there is no relationship between average SAT scores and the number of undergraduates enrolled at a college, r(96) = 0.134, p > .05.

### Cost and Debt

To see the relationship between *average cost of attendance (tuition and room and board less all grant aid)* and *median debt of graduates*, I plotted the two variabes using the plot function.

## Plot of Cost and Debt  
plot(sc$costt4\_a, sc$debt\_mdn,main="Cost and Debt", xlab="Cost", ylab="Debt", pch=19)

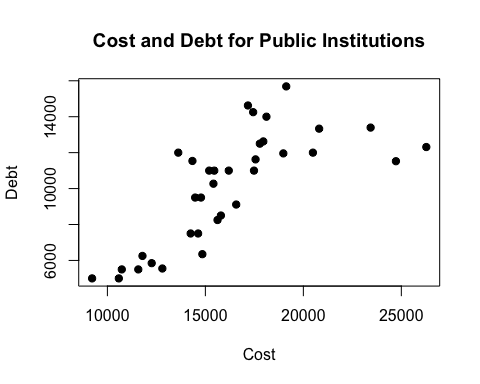


It’s apparent from the plot that a positive relationship exists between a college’s cost and the debt incurred by its students. This is not surprising.

### Cost and Debt by Control of Institution

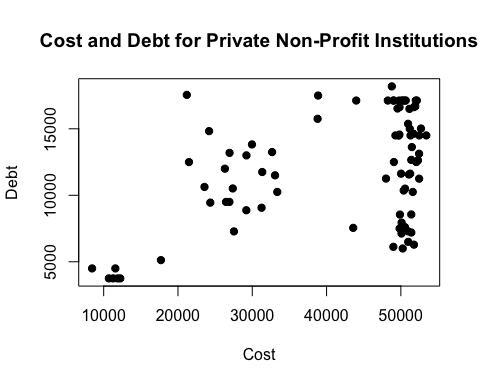
To plot the cost and debt relationship by control of institution, I first passed the sc data set through a filter for control of instituion. I then assigned the filtered results a name \*(public, private\_np, and private\_fp) before executing a plot function. For example, to plot the cost/debt relationship for public institutions, I set the control paraemeter equal to 1.

## Plot of cost and debt for public institutions  
public <- sc%>%filter(control==1)  
  
plot(public$costt4\_a, public$debt\_mdn, main="Cost and Debt for Public Institutions", xlab="Cost", ylab="Debt", pch=19)



To plot the cost/debt relationship for private non-profit institutions, I set the control paraemeter equal to 2.

## Plot of cost and debt for private non-profit institutions  
private\_np <- sc%>%filter(control==2)  
  
plot(private\_np$costt4\_a, private\_np$debt\_mdn,main="Cost and Debt for Private Non-Profit Institutions", xlab="Cost", ylab="Debt", pch=19)



Finally, to plot the cost/debt relationship for private for-profit institutions, I set the control paraemeter equal to 3.

## Plot of cost and debt for private for-profit institutions  
private\_fp <- sc%>%filter(control==3)  
  
plot(private\_fp$costt4\_a, private\_fp$debt\_mdn,main="Cost and Debt for Private For-Profit Institutions", xlab="Cost", ylab="Debt", pch=19)

