STAT GU4206/5206 Midterm

Christine Chong cc4190

June 8, 2017

The STAT GU4206/5206 midterm is open notes, open book(s), and online resources are allowed. Students are **not** allowed to communicate with any other people during the midterm with the exception of the GU4206/5206 instructor. When you are finished with the midterm, please upload both the .pdf and .Rmd files on Canvas.

For the entire midterm we consider the **Auto** dataset taken from the *Introduction to Statistical Learning* package. Before starting the midterm, make sure the package **ISLR** is installed on your laptop.

```
#install.packages("ISLR")
library(ISLR)
```

Problem 1: Basic Operations

1.i)

Display the first 3 rows of the **Auto** dataset:

```
head(Auto,3)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
```

1.ii)

How many rows are in this dataset?

```
nrow(Auto)
```

```
## [1] 392
```

Problem 2: Regression and the Bootstrap

2.i)

Consider extracting only the rows corresponding to 6 cylinder cars. Also consider running a linear regression on a car's acceleration versus its weight. The working filtering and linear regression code is displayed below.

```r
Auto.6.cyl <- Auto[Auto$cylinders==6,]
beta.hats <- coef(lm(acceleration~weight,data=Auto.6.cyl))
beta.hats
```

```
## (Intercept)      weight
## 4.368123570 0.003711944
```

Identify the estimated slope and intercept of the above linear model.

Solution: The estimated slop of the model should be the weight which is around 0.0037. The estimated intercept of the model should be 4.38.

2.ii)

Now suppose as researchers, we want to infer upon the true slope relating a six cylinder car's acceleration versus its weight. Also suppose that we do not want to make strong assumptions on the errors of the linear regression model; hence we will perform a bootstrap procedure. Below is *almost complete* working code that runs a bootstrap procedure for the slope. Fill in the one missing line of **R** code and make sure to uncomment each line below. Run the bootstrap procedure after filling in the missing line.

```r
 set.seed(0)
 B <- 1000
 n <- nrow(Auto.6.cyl)
 slopes.boot <- rep(NA,B)
 for (b in 1:B) {
   sample.boot <- sample(1:n, n, replace = TRUE )
   regression.boot <- lm(acceleration~weight,data=Auto.6.cyl[sample.boot,])
   slopes.boot[b] <- coef(regression.boot)[2]

 }
 slope.hat <- beta.hats[2]
 LL <- 2*slope.hat-quantile(slopes.boot,.975)
 UL <- 2*slope.hat-quantile(slopes.boot,.025)
 c(LL,UL)
```

```
##      weight      weight
## 0.002748981 0.004727104
```

2.iii)

At 5% significance, is a six cylinder car's acceleration statistically related to its weight? Support your answer using the computed bootstrap interval **c(LL,UL)**.

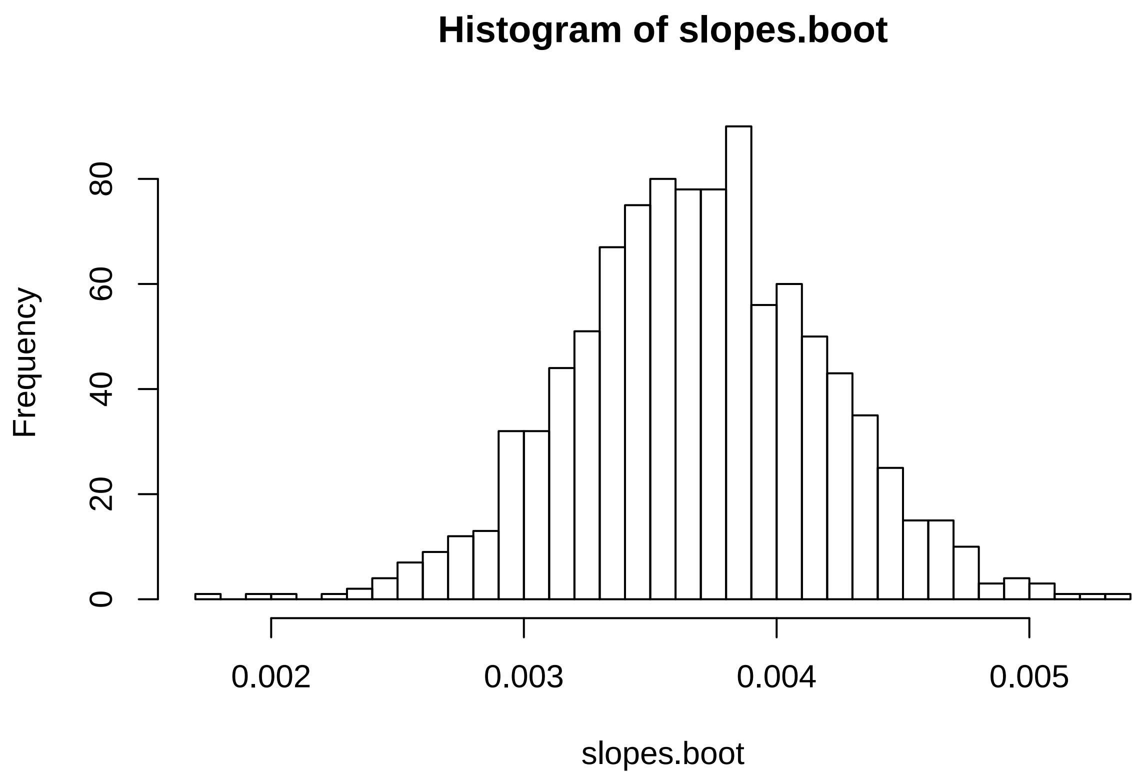
Solution:

Due to the value at 5% significance being not 0, it is safe to say that a six cylinder's car accleration is stastiscally related. If the number was closer to 1, it would have more of a significant relation.

2.iv)

Create a histogram of the bootstrapped slope estimates. Make sure to label the histogram appropriately and use 30 breaks for the bins.

```r
hist(slopes.boot, breaks = 30)
```



Problem 3: Subseting

The original **Auto** datafrme consists of cars with 3,4,5,6 and 8 cylinders. Create a new dataframe named **Auto.new** that consists of only the cars with 4,6 and 8 cylinders. Check that the number of rows in this new dataframe is equal to 385.

```r
newdata <- rbind(Auto[Auto$cylinders == "4",], Auto[Auto$cylinders =="6",], Auto[Auto$cylinders =="8",])
Auto.new <- data.frame(newdata)
dim(Auto.new)
```

```
## [1] 385   9
```

Problem 4: Character Srings and Regular Expressions

4.i)

Look at the first few cases of the variable **name**.

```r
head(Auto$name)
```

```
## [1] chevrolet chevelle malibu buick skylark 320
## [3] plymouth satellite        amc rebel sst
```

```
## [5] ford torino               ford galaxie 500
## 304 Levels: amc ambassador brougham ... vw rabbit custom
```

Notice that the first word in each string is the car's company, i.e., chevrolet, buick, toyota, etc… Append a new variable on the **Auto.new** dataframe named **company** that displays the company of each car. For example, if the **name** of the car is "chevrolet chevelle malibu", then the car's company should be "chevrolet". Show the first three observations in this new dataset.

Note: You might have to convert the factor variable back into a character variable.

```r
max = nrow(Auto.new)
Auto.new$name <- as.character(Auto.new$name)
for(i in 1:max){
  new <- strsplit(Auto.new$name[i], split = " ")
  unlistnew <- unlist(new)
  titlecompany <- unlistnew[1]
  titlecompany
  Auto.new$company[i] <- titlecompany
}
head(Auto.new, 3)
```

```
##    mpg cylinders displacement horsepower weight acceleration year origin
## 15  24         4          113         95   2372         15.0   70      3
## 19  27         4           97         88   2130         14.5   70      3
## 20  26         4           97         46   1835         20.5   70      2
##                            name    company
## 15        toyota corona mark ii     toyota
## 19                 datsun pl510     datsun
## 20 volkswagen 1131 deluxe sedan volkswagen
```

4.ii)

When the experimenter was recording the data, he entered a few typos for the car's company names, i.e., one case shows "toyouta" and another case shows "vokswagen". Fix these two typos in the **Auto.new** dataframe by using the **grep** function to find the location of the typos and then assigning new strings to these elements. After fixing the typos, create a table of the variable **company**.

```r
loc1<-grep("toyouta", Auto.new$name)
Auto.new$name[loc1] <- "toyota corona mark ii(sw)"
loc2<-grep("vokswagen", Auto.new$name)
Auto.new$name[loc2] <- "volkswagen rabbit"
Auto.new$name[loc1]
```

```
## [1] "toyota corona mark ii(sw)"
```

```r
Auto.new$name[loc2]
```

```
## [1] "volkswagen rabbit"
```

Problem 5: The Apply Family

5.i)

Using the appropriate apply function, compute the maximum **horsepower** per **company**. Also sort this output.

```r
horsemax <- tapply(Auto.new$horsepower, Auto.new$company, max)
horsemax
```

```
##           amc          audi           bmw         buick      cadillac 
##           190            95           113           225           180 
##         capri     chevroelt     chevrolet         chevy      chrysler 
##            92           105           220           200           215 
##        datsun         dodge          fiat          ford            hi 
##           132           210            90           215           193 
##         honda         maxda         mazda mercedes-benz       mercury 
##            97            65            75           120           208 
##        nissan    oldsmobile          opel       peugeot      plymouth 
##            88           180            90           133           215 
##       pontiac       renault          saab        subaru        toyota 
##           230            83           115            93           122 
##       toyouta       triumph     vokswagen    volkswagen         volvo 
##            97            88            62            78           125 
##            vw 
##            76
```

5.ii)

Using the appropriate apply function, compute the average value of quantitative variables **mpg**, **displacement**, **horsepower**, **weight** and **acceleration**. To save some time I provided the vector of character strings in the below code chunk.

```r
variables <- c("mpg","displacement","horsepower","weight","acceleration")
apply(Auto.new[variables],2,mean)
```

```
##          mpg displacement   horsepower       weight acceleration 
##     23.44545    196.06364    104.69610   2982.62078     15.54104
```

Problem 6: R Base Graphics

Construct a base **R** plot that shows a car's acceleration (Y) versus a its weight (X) split by the number of cylinders in the car (4,6, and 8). Note that you should be using the **Auto.new** dataset. For full credit, create the scatter plot and split the data up by different colors to represent the number of cylinders. Also create a legend and label the plot appropriately.

For extra credit, plot regression lines for each subgroup, i.e., plot 3 least squares lines: one line for 4 cylinders, one line for 6 cylinders and one line for 8 cylinders.

```r
plot(Auto.new$weight, Auto.new$acceleration, col= factor(Auto.new$cylinders))
legend("bottomright", legend = levels(factor(Auto.new$cylinders)), fill = unique(factor(Auto.new$cylinde
cylinders <- levels(factor(Auto.new$cylinders))
col_counter <- 1
for (i in cylinders) {
  this_cylinder    <- Auto.new$cylinders == i
  this_data   <- Auto.new[this_cylinder, ]
  this_lm     <- lm((this_data$acceleration)
                    ~(this_data$weight))
  abline(this_lm, col = col_counter)
```

```
  col_counter <- col_counter + 1
}
```