**I421: Applications in Data Mining**

**Homework 4**

**Total Points: 47 Points**

Practice 1

Find the College.csv in Data Set folder on Canvas; we will use this dataset to perform some practice.

a. Use read.csv() to load the data into the R dataframe with name college (2 points)

b. Find the how many number of data objects in this dataframe (2 points)

777 rows, 19 columns

c. Use fix() and View() function to quickly check the dataframe that you just loaded. And use rownames(college) to check the names in row direction (3 points)

d. Try the following code:

rownames (college )=college [,1]

View (college )

What do you find after these 2 lines of codes? (2 points)

The repeated first column is removed.

e. Let’s continue to modify the dataframe to be a good look

college =college [,-1]

View(college)

What do you find after these 2 lines of codes? (2 points)

Column 1 is set as the index column and no longer has the variable X

f. Use the summary() function to generate a numerical summary of the variable in the data set, draw a short statement of what you find. (3 points)

* There are more than twice as many private universitied than public ones
* A median of 1558 apps are presented, a median 2019 of them are accepted, and a median of 434 students end up enrolling.
* The means for this data are around twice as much as the median, meaning that the data is significantly skewed to the right.
* There is also data on the undergrads, outstate, room and board, books, personal, PHD students, etc. Some of these would require some metadata to fully understand what they mean.

g. Create a new qualitative variable with name Elite, we are going to do some study for attribute Top10perc, We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50%.

Elite =rep ("No",nrow(college ))

Elite [college$Top10perc >50]=" Yes"

Elite =as.factor (Elite)

college =data.frame(college ,Elite)

(Here, as.factor() function converts quantitative variables into qualitative as.factor() variables.)

Use summary() to see how many elite universities in datafram (4 points)

h. Use hist() function to plot the histogram for attributes: Top10perc, Top25perc, F.Undergrad, Outstate (4 points)

Practice 2

Find the Auto.csv in Data Set folder on Canvas; we will use this dataset to perform some practice.

a. Read the data file into a dataframe. (2 points)

b. Find the dimension of the dataframe, how many data objects and attributes are in this data frame (3 points)

397 rows, 9 columns

Now use the following line to re-load the Auto.csv to the dataframe

auto = read.csv(“Auto.csv”, header = TRUE, na.string = “?”)

Here, na.string defines that any time R sees a particular character or set of characters (such as a question mark), it should be treated as a missing element of the data.

c. Use R functions to remove data with missing values (2 points)

d. What is the range of each quantitative attribute? You can answer this using the range() function. (4 points)

> mpgRange <- range(auto$mpg)

> mpgRange

[1] 9.0 46.6

> cylindersRange <- range(auto$cylinders)

> cylindersRange

[1] 3 8

> dispRange <- range(auto$displacement)

> dispRange

[1] 68 455

> hpRange <- range(auto$horsepower)

> hpRange

[1] 46 230

> weightRange <- range(auto$weight)

> weightRange

[1] 1613 5140

> accRange <- range(auto$acceleration)

> accRange

[1] 8.0 24.8

> yearRange <- range(auto$year)

> yearRange

[1] 70 82

e. What is the mean and standard deviation of each quantitative attribute? (4 points)

> mpgMean <- mean(auto$mpg)

> mpgMean

[1] 23.44592

> mpgSD <- sd(auto$mpg)

> mpgSD

[1] 7.805007

>

> cylindersMean <- mean(auto$cylinders)

> cylindersMean

[1] 5.471939

> cylindersSD <- sd(auto$cylinders)

> cylindersSD

[1] 1.705783

>

> dispMean <- mean(auto$displacement)

> dispMean

[1] 194.412

> dispSD <- sd(auto$displacement)

> dispSD

[1] 104.644

>

> hpMean <- mean(auto$horsepower)

> hpMean

[1] 104.4694

> hpSD <- sd(auto$horsepower)

> hpSD

[1] 38.49116

>

> weightMean <- mean(auto$weight)

> weightMean

[1] 2977.584

> weightSD <- sd(auto$weight)

> weightSD

[1] 849.4026

>

> accMean <- mean(auto$acceleration)

> accMean

[1] 15.54133

> accSD <- sd(auto$acceleration)

> accSD

[1] 2.758864

>

> meanYear <- mean(auto$year)

> meanYear

[1] 75.97959

> yearSD <- sd(auto$year)

> yearSD

[1] 3.683737

f. Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each attribute in the subset of the data that remains? (4 points)

> mpgRangeCleaned <- range(auto$mpg)

> mpgRangeCleaned

[1] 9.0 46.6

> mpgMeanCleaned <- mean(autosCleaned$mpg)

> mpgMeanCleaned

[1] 24.40443

> mpgSDCleaned <- sd(autosCleaned$mpg)

> mpgSDCleaned

[1] 7.867283

>

> cylindersRangeCleaned <- range(autosCleaned$cylinders)

> cylindersRangeCleaned

[1] 3 8

> cylindersMeanCleaned <- mean(autosCleaned$cylinders)

> cylindersMeanCleaned

[1] 5.373418

> cylindersSDCleaned <- sd(autosCleaned$cylinders)

> cylindersSDCleaned

[1] 1.654179

>

> dispRangeCleaned <- range(autosCleaned$displacement)

> dispRangeCleaned

[1] 68 455

> dispMeanCleaned <- mean(autosCleaned$displacement)

> dispMeanCleaned

[1] 187.2405

> dispSDCleaned <- sd(autosCleaned$displacement)

> dispSDCleaned

[1] 99.67837

>

> hpRangeCleaned <- range(autosCleaned$horsepower)

> hpRangeCleaned

[1] 46 230

> hpMeanCleaned <- mean(autosCleaned$horsepower)

> hpMeanCleaned

[1] 100.7215

> hpSDCleaned <- sd(autosCleaned$horsepower)

> hpSDCleaned

[1] 35.70885

>

> weightRangeCleaned <- range(autosCleaned$weight)

> weightRangeCleaned

[1] 1649 4997

> weightMeanCleaned <- mean(autosCleaned$weight)

> weightMeanCleaned

[1] 2935.972

> weightSDCleaned <- sd(autosCleaned$weight)

> weightSDCleaned

[1] 811.3002

>

> accRangeCleaned <- range(autosCleaned$acceleration)

> accRangeCleaned

[1] 8.5 24.8

> accMeanCleaned <- mean(autosCleaned$acceleration)

> accMeanCleaned

[1] 15.7269

> accSDCleaned <- sd(autosCleaned$acceleration)

> accSDCleaned

[1] 2.693721

>

> yearRangeCleaned <- range(autosCleaned$year)

> yearRangeCleaned

[1] 70 82

> meanYearCleaned <- mean(autosCleaned$year)

> meanYearCleaned

[1] 77.14557

> yearSDCleaned <- sd(autosCleaned$year)

> yearSDCleaned

[1] 3.106217

Question 3

Create a binary operator %&% that will concatenate two strings as a single string with a space as a seperator. Write an operator function in which the quotation marks around the terms are optional.

The following output is for quoted and/or unquoted arguments. (6 points)

> Hello %&% World

[1] “Hello World”

> “Hello” %&% “World”

[1] “Hello World”