**MIS 6356 Exam 1**

1. **Data Exploration (20 Points)**
2. **Import the “Universities-Exam1.csv” file to R Studio and write R code to produce summary statistics of all numeric variables. For each numeric variable, include the following.**

**Variable name**

**Minimum**

**Maximum**

**Median**

**Mean**

**Standard Deviation**

**Number of Missing Values**

**Paste a screenshot of your summary statistics below.**

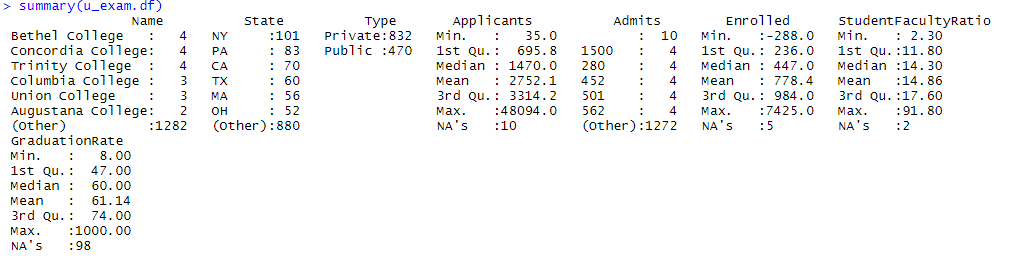
#load universities-exam1.csv from working directory

u\_exam.df <- read.csv("Universities-Exam1.csv")

View(u\_exam.df)

#Basic summary statistics

summary(u\_exam.df)



#coerce dataframe to a list

u\_exam\_list <- as.list(u\_exam.df)

str(u\_exam\_list)

#select numeric columns from the dataset

u\_exam\_num.df <- u\_exam.df[,c(4,5,6,7,8)]

#select numeric columns from the dataset

View(u\_exam\_num.df)

str(u\_exam\_num.df)

#converting admits factor to numeric valued column

u\_exam\_num.df$Admits <- as.numeric(as.vector(u\_exam\_num.df$Admits))

str(u\_exam\_num.df)

#getting mean, median, missing values, minimum, maximum, variance, standard deviation using sapply function

u\_exam\_stats\_df <- data.frame( mean=sapply(u\_exam\_num.df,na.rm=TRUE, mean),

median=sapply(u\_exam\_num.df,na.rm=TRUE, median),

sd=sapply(u\_exam\_num.df,na.rm=TRUE, sd),

variance=sapply(u\_exam\_num.df, na.rm=TRUE,var),

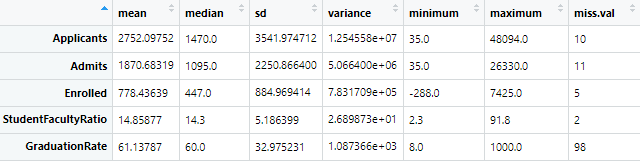
minimum=sapply(u\_exam\_num.df, na.rm=TRUE,min),

maximum=sapply(u\_exam\_num.df, na.rm=TRUE,max),

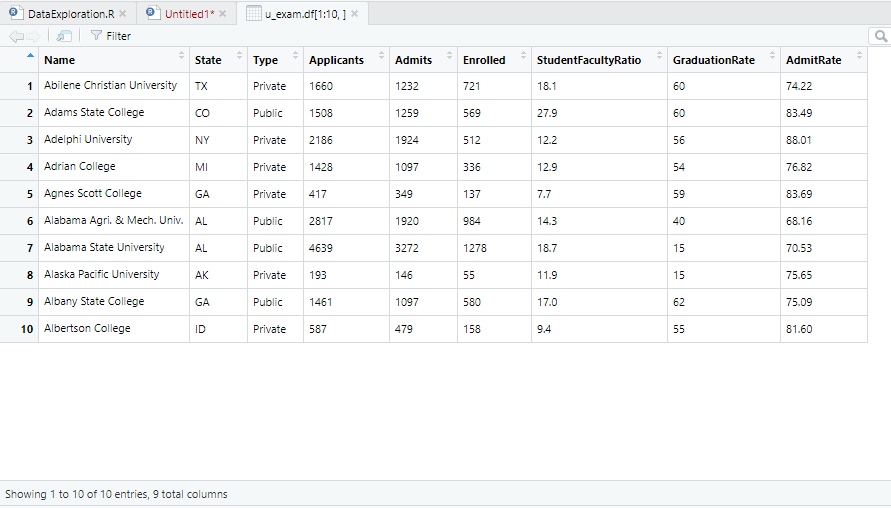
miss.val=sapply(u\_exam\_num.df,function(x)

sum(length(which(is.na(x))))))

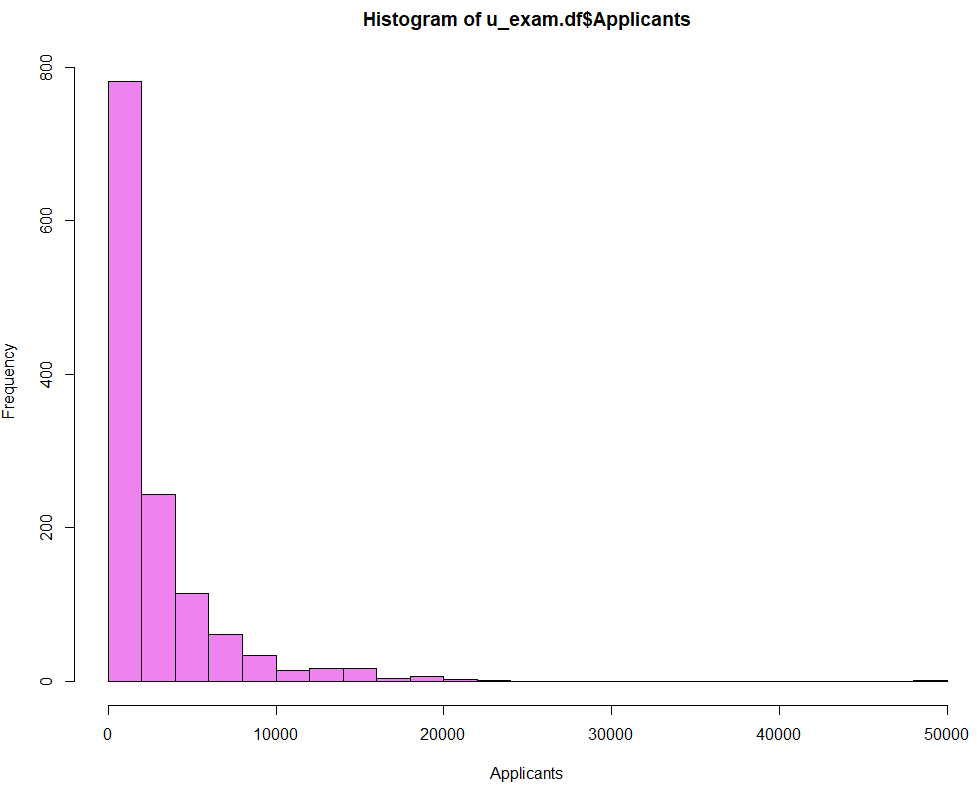
View(u\_exam\_stats\_df)



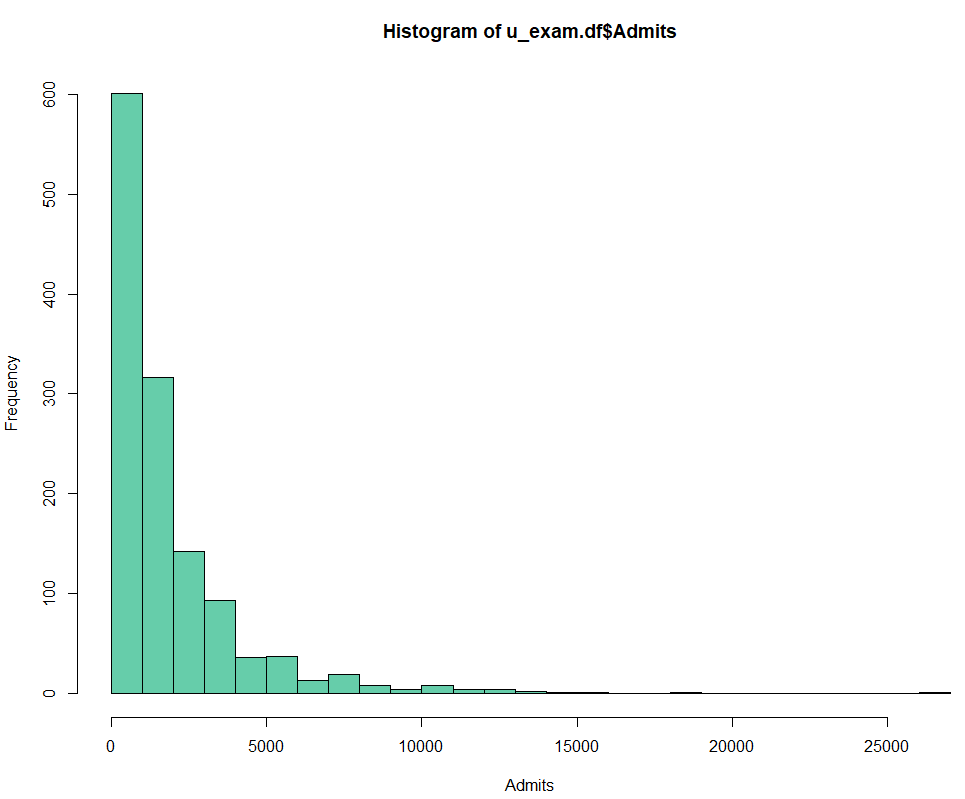
1. **Create a new column called “AdmitRate” and add it to the data set as the last column. Admit rate should be calculated by dividing “Admits” by “Applicants” to get the percent of students admitted. Paste a screen shot of the first 10 rows of the dataset with the new column below.**



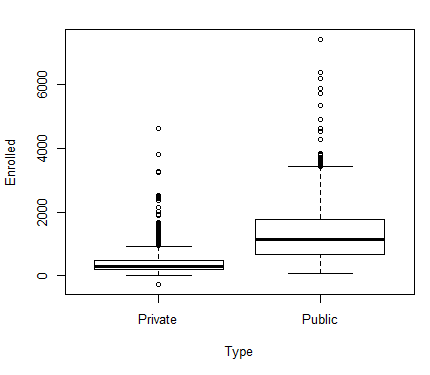
1. **Create a histogram of the Applicants variable. Paste a screen shot below.**



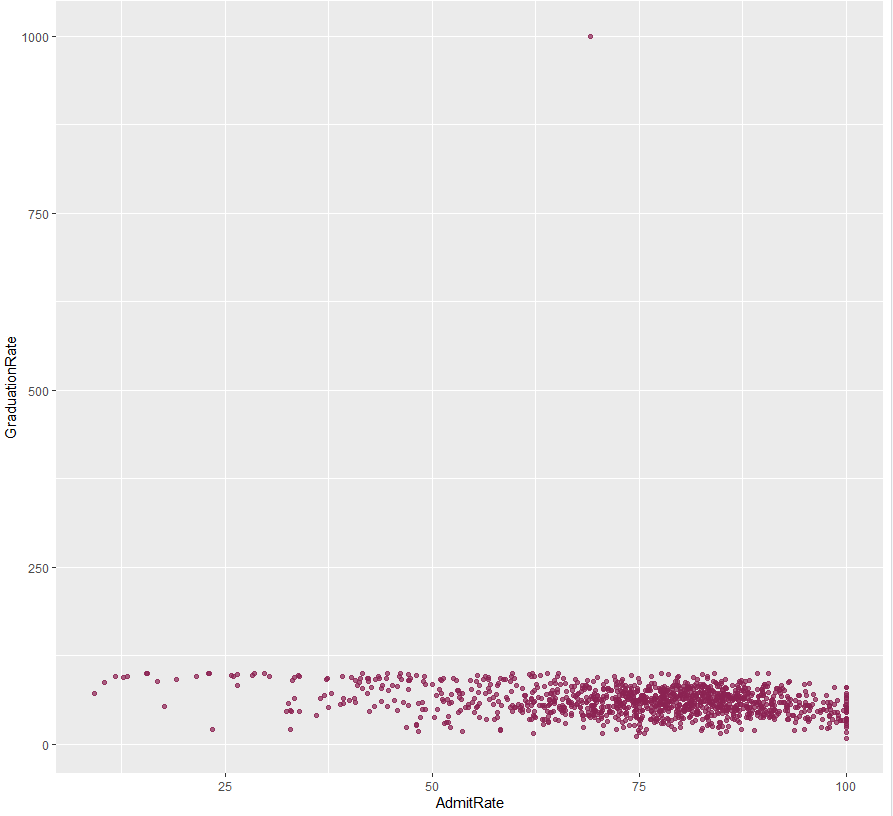
1. **Create a histogram of Admits. Paste a screenshot below.**



1. **Create a side by side Box Plot for Enrolled using Type as the by variable.**



1. **Create a scatter plot of AdmitRate and GraduationRate**



1. **Find three errors in the data provided.**

The following errors have been noted in the dataset

1. Enrolled has a negative value of -288
2. Admits has a special character ''
3. Entire dataset has 125 empty values
4. GraduationRate has a value of single highest value of 1000
5. **Paste a copy of your R code below.**

#load universities-exam1.csv from working directory

u\_exam.df <- read.csv("Universities-Exam1.csv")

View(u\_exam.df)

str(u\_exam.df)

#Basic summary statistics

summary(u\_exam.df)

#change Admits which is factor to numeric

u\_exam.df$Admits <- as.numeric(as.vector(u\_exam.df$Admits))

str(u\_exam.df)

#coerce dataframe to a list

u\_exam\_list <- as.list(u\_exam.df)

str(u\_exam\_list)

u\_exam\_num.df <- u\_exam.df[,c(4,5,6,7,8)]

View(u\_exam\_num.df)

str(u\_exam\_num.df)

u\_exam\_stats\_df <- data.frame( mean=sapply(u\_exam\_num.df,na.rm=TRUE, mean),

median=sapply(u\_exam\_num.df,na.rm=TRUE, median),

sd=sapply(u\_exam\_num.df,na.rm=TRUE, sd),

variance=sapply(u\_exam\_num.df, na.rm=TRUE,var),

minimum=sapply(u\_exam\_num.df, na.rm=TRUE,min),

maximum=sapply(u\_exam\_num.df, na.rm=TRUE,max),

miss.val=sapply(u\_exam\_num.df,function(x)

sum(length(which(is.na(x))))))

View(u\_exam\_stats\_df)

u\_exam.df$AdmitRate <- round((u\_exam.df$Admits / u\_exam.df$Applicants) \* 100,2)

View(u\_exam.df[1:10,])

## histogram of MEDV

hist(u\_exam.df$Applicants, xlab = "Applicants", col = "violet" , breaks= 30)

## histogram of MEDV

hist(u\_exam.df$Admits, xlab = "Admits", col = "mediumaquamarine" , breaks= 30)

#side by side box plot

par(mfcol = c(2, 2))

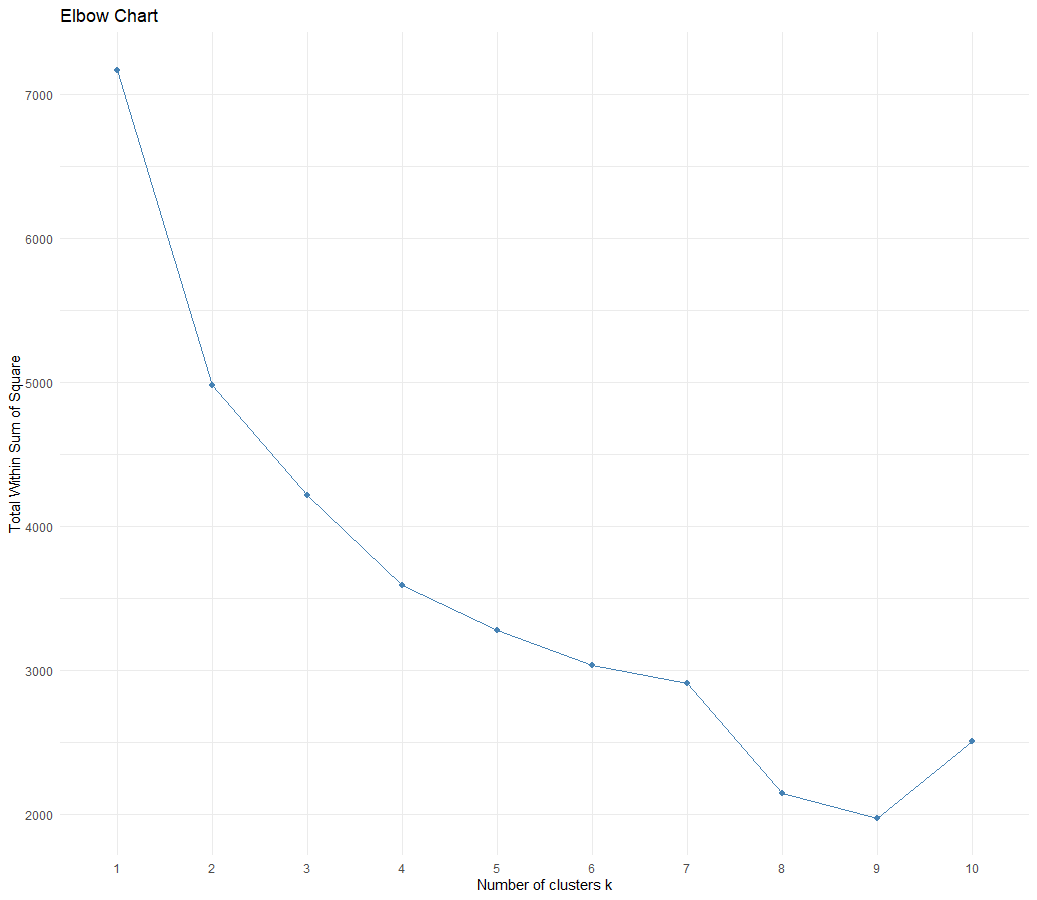
boxplot(u\_exam.df$Enrolled ~ u\_exam.df$Type, xlab = "Type", ylab = "Enrolled")

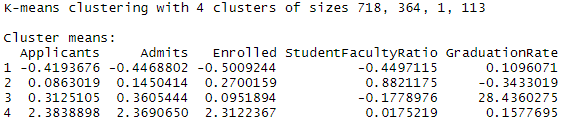
#scatter plot

library(ggplot2)

ggplot(u\_exam.df) + geom\_point(aes(x = AdmitRate, y = GraduationRate), colour = "violetred4", alpha = 0.7)

1. **Cluster Analysis (20 points)**
2. **Perform a k-means cluster analysis on the Universities data set used in part 1. Use k = 4, standardize the data, and remove all rows with missing values prior to performing the cluster analysis. Paste a screenshot of the means of the cluster centers below.**

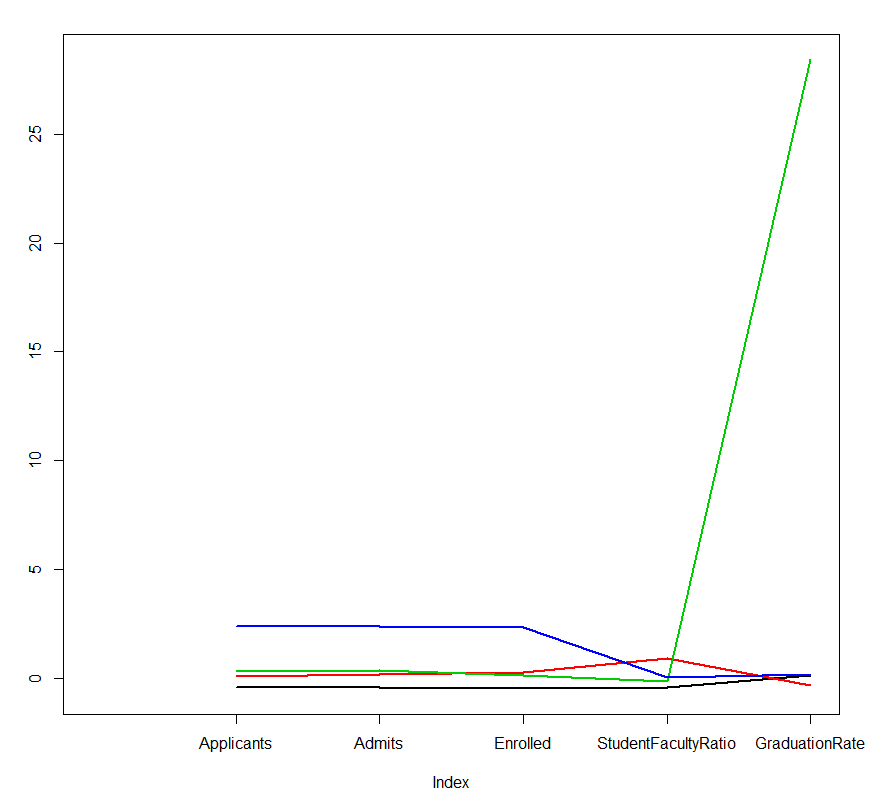


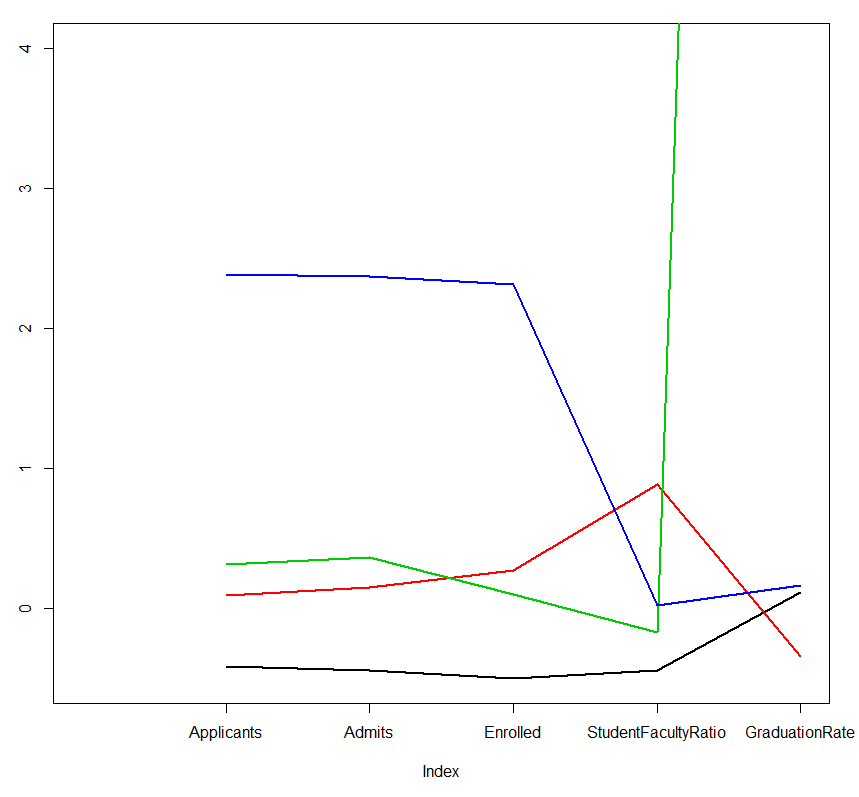


1. **Create a line plot of the cluster centers similar to the following. Paste a screenshot of your plot below.**

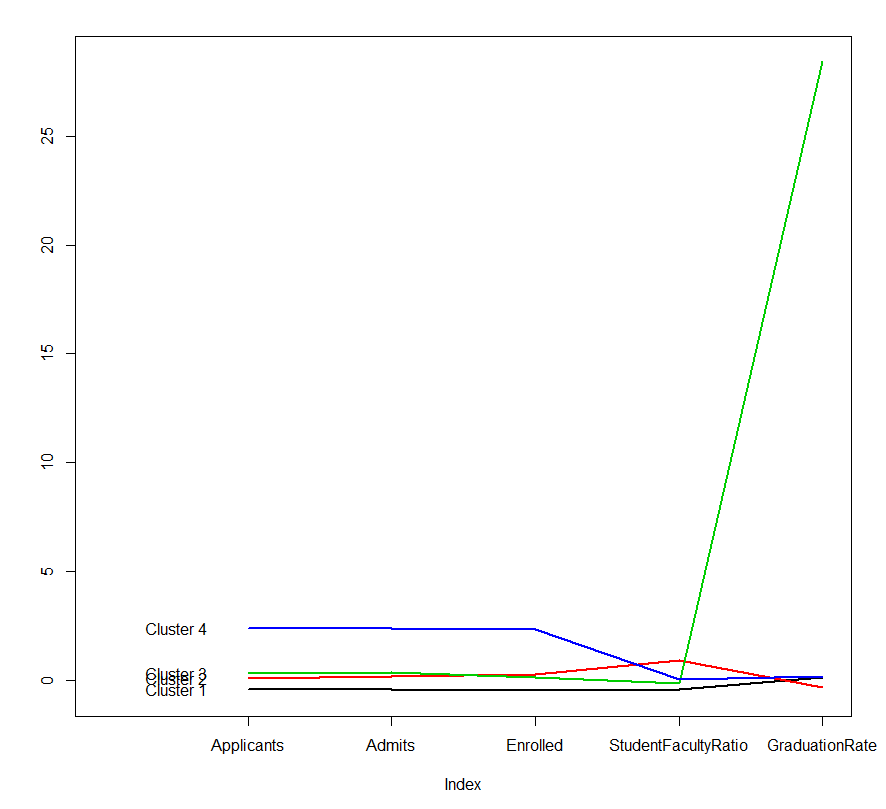
A close up of a map

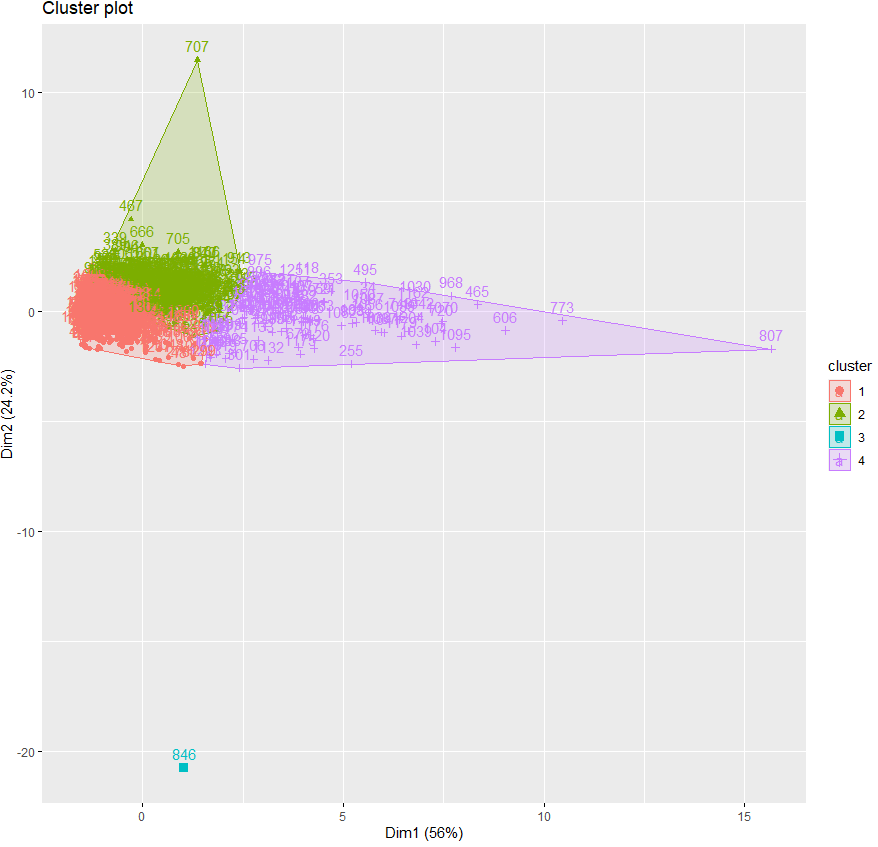
Description automatically generated

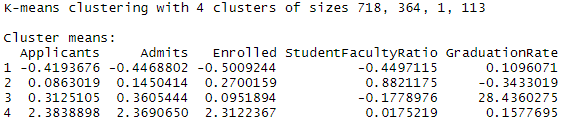




1. **Interpret the clusters that have been created and create a name for each of the clusters.**







* Cluster 1 has a highest density of colleges of 718 with low admits, applicants and enrolled. These data points very close together and the graduation rate is good. This cluster has below average values for applicants, admits , enrolled and student faculty ratio
* In Cluster 2, the colleges have below average graduation rate and it dense with 364 count and has one drastic outlier
* Cluster 3 has the highest graduation rate of 1000 and is a single data point cluster as it doesn’t match with other clusters because of high graduation rate and low student faculty ratio
* Cluster 4 has colleges with high mean of applicants, admits and enrolled with more outliers spreading out

1. **Paste a copy your R code below.**

#load universities-exam1 dataset

u\_exam.df<- read.csv("Universities-Exam1.csv")

#View(u\_exam.df)

u\_exam.df$Admits <- as.numeric(as.vector(u\_exam.df$Admits))

#str(u\_exam.df)

#u\_exam.df$AdmitRate <- round((u\_exam.df$Admits / u\_exam.df$Applicants) \* 100,2)

u\_exam\_clean.df <- na.omit(u\_exam.df)

#View(u\_exam\_clean.df)

u\_exam\_numeric.df <- u\_exam\_clean.df[,c(4:8)]

#View(u\_exam\_numeric.df)

u\_exam\_norm.df <- scale(u\_exam\_numeric.df,center=TRUE, scale= TRUE)

#View()

#caculcate normalised distance based on all variables

distance.norm <- dist(u\_exam\_norm.df, method="euclidean")

#print(distance.norm)

mat <- as.matrix(distance.norm)

#View(mat)

#elbow chart

#install.packages("factoextra")

library(factoextra)

fviz\_nbclust(u\_exam\_norm.df, kmeans, method="wss")+theme\_minimal()+ ggtitle("Elbow Chart")

#kmeans with k=4

set.seed(2)

kmu4 <- kmeans(u\_exam\_norm.df, 4, nstart = 100)

print(kmu4)

# show cluster membership

kmu4$cluster

cluster.df <- data.frame(kmu4$cluster)

View(kmu4$centers)

# centroids

kmu4$centers

#plot an empty scatter plot - centroid plot

plot(c(0),xaxt='n', ylab="", type="l", ylim=c(min(kmu4$centers),max(kmu4$centers)),xlim=c(0,5))

#label x-axis

axis(1, at=c(1:5) , labels=c(colnames(u\_exam\_norm.df)))

#draw lines

for (i in c(1:4))

lines(kmu4$centers[i,], lwd = 2, col = i)

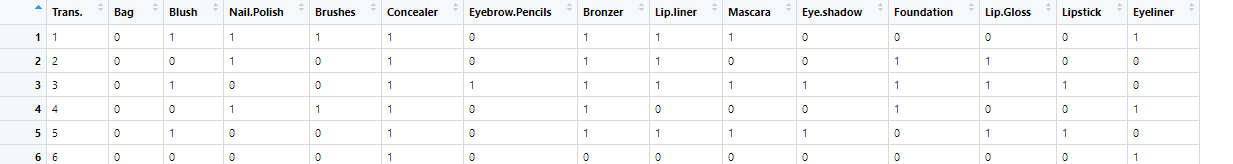
#name

text(x = 0.5, y = kmu4$centers[, 1], labels = paste("Cluster", c(1:4)))

1. **Association Rules (20 points)**
2. **Import the Cosmetics.csv file into R Studio and browse the dataset to understand it. What does each row represent?**

The dataset represents the transactions processed in a cosmetics shop. It has 15 columns denoting the 14 products sold and the transaction number in the 1st column. The dataset has 880 observations.

The dataset has binary representation of the products bought in each transaction with 1 representing the product been bought and 0 representing the product not been bought.



The items sold which are shown by each column :

Bag

Blush

Nail Polish

Brushes

Concealer

Eyebrow Pencils

Bronzer

Lip Liner

Mascara

Eye Shadow

Foundation

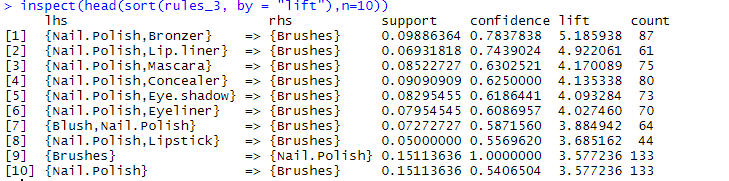
Lip Gloss

Lipstick

Eyeliner

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| It is in data frame structure |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Perform an association rules analysis with a minimum support of 5%, a minimum threshold confidence of 50%, and a maximum rule length of 3 items. Paste a screenshot of the top 10 rules in descending order by lift below.**



1. **Explain the first rule.**

If { Nail polish and Bronzer} => Then { Brushes }

*The first rule states that If { Nail polish and Bronzer} are bought together, Then there is a good probability that { Brushes } will also be bought.*

The interpretations of this rule can be done with the help of the support, confidence ,lift and values provided along with each rule.

1. The probability that the customer bought Nail Polish, Bronzer and Brushes together is 9% which is given by the number of transactions done with {Nail Polish, Bronzer,Brushes} of all the transactions done. This shows the relevance of the rule with respect to all the transactions done.
2. The probability that the customer bought {Brushes} given that she has already bought { Nail Polish, Bronzer } is given the confident value of 78%. This provides the conditional probability of buying Brushes given that Nail Polish and Bronzer are already purchased.
3. The rule can be seen applicable in 87 count of the transactions done of the total 880 transactions.
4. The lift of 5.18 implies a strong association between these three items that are bought.
5. **Provide the owner of the store selling cosmetics with a recommendation on how the first rule could be used to improve his or her business.**

Dear Store Management,

I had analyzed 880 transaction records coming through your store and found a set of patterns emerge from the transaction records when I ran the apriori association function on it with a given minimum support level of 5% and confidence level of 50%. The association function provides a set of rules that identifies items that have a co dependence of being bought together during a transaction. I recommend the very first rule in the set of the rules that had emerged, to be implemented in your store to boost sales as it has the highest lift value associated with it. It has been seen that customers in a cosmetic store tend to buy Brushes along with Nail Polish and Bronzer. This rule can be substantiated with the occurrence of 87 times in the set of transactions I had analyzed. Also, the probability that customers bought Bronzers given that they had already purchased Nail Polish and Bronzer is high at 78%. This implies that almost ¾ of the transactions having Nail Polish and Bronzer have a Brush being bought. I recommend you target and sell Brush to these customers who have already put into cart Nail Polish and Bronzer. You can achieve this by following the following marketing strategies:

* Bundling these items together
* Placing these items close to each other in the aisles
* A/B testing this rule to see the percentage boost in sales
* Providing offers in these bundled products.

1. **Paste a copy of the R code below.**

#load required libraries

library(Matrix)

library(arules)

#import file

cosmetics.df <- read.csv("Cosmetics-Exam1.csv")

str(cosmetics.df)

class(cosmetics.df)

View(cosmetics.df)

# create a binary incidence matrix

# remove variables that don't contain purchase information

count.cosmetics.df <- cosmetics.df[, c(2:15)]

# change purchase columns to binary yes/no

incid.cosmetics.num <- ifelse(count.cosmetics.df > 0, 1, 0)

str(incid.cosmetics.num)

#coerce the binary incidence matrix/array into a 'transactions' object

cosmetics.trans <- as(incid.cosmetics.num, "transactions")

# run apriori function

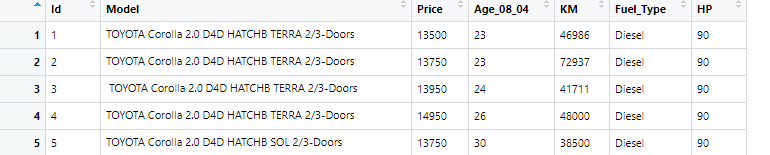
rules\_3 <- apriori(cosmetics.trans,

parameter = list(supp= 0.05, conf = 0.5, maxlen = 3, target = "rules"))

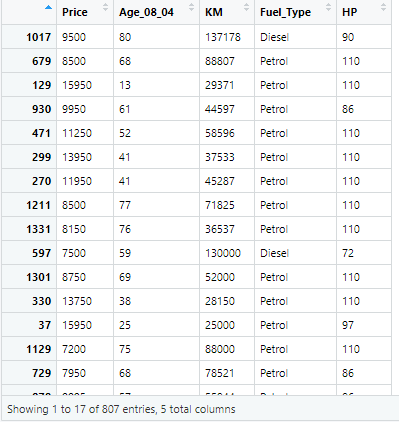
# inspect and sort the 10 rows with high lift

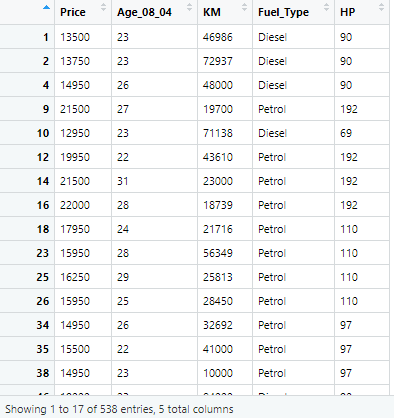
inspect(head(sort(rules\_3, by = "lift"),n=10))

1. **Regression Analysis (20 points)**
2. **Import the ToyotaCorolla-Exam1.csv file to R Studio.**

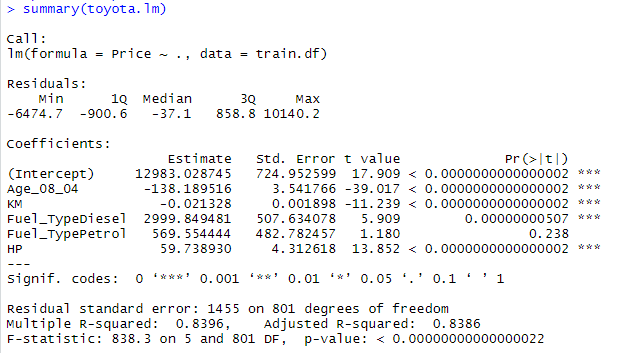


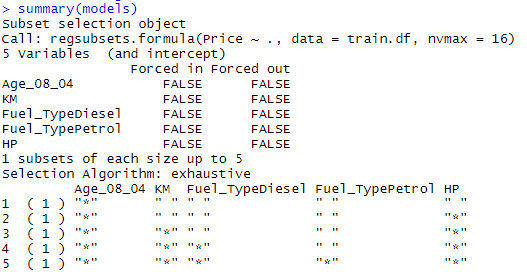
1. **Randomly select 60% of the cars and assign them to a training partition. The remaining 40% of cars should be assigned to a validation partition.**

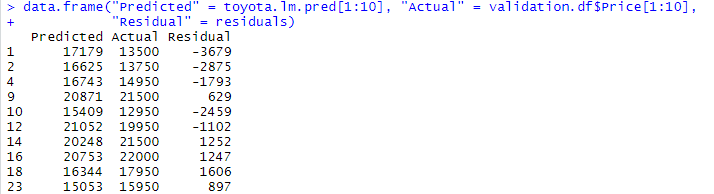


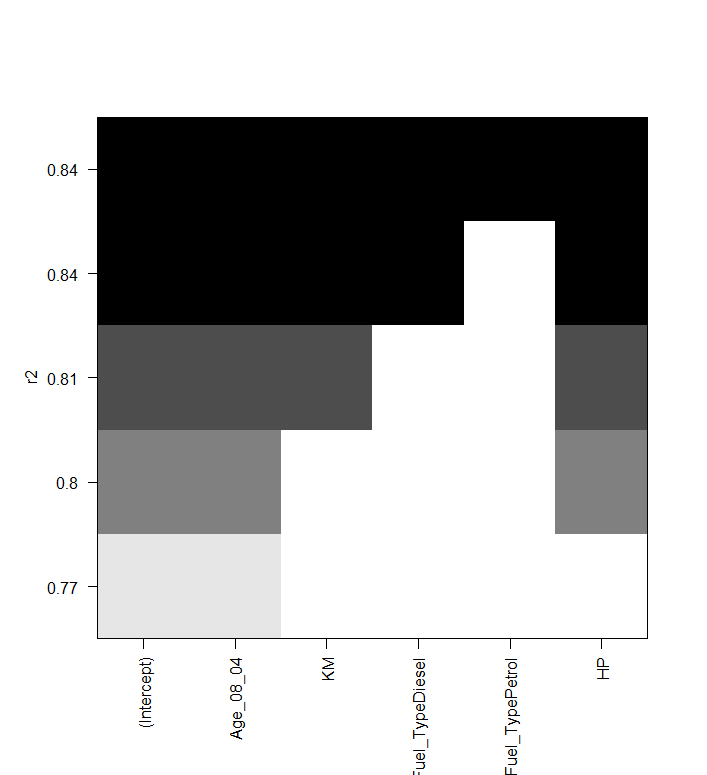


1. **Fit a linear regression model on the training partition using OLS with Price as the dependent variable and Age\_08\_04, KM, Fuel\_Type and HP as the predictors.**

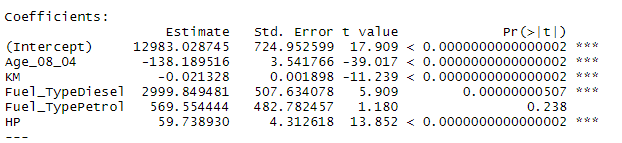








1. **Write the equation of the fitted model below.**



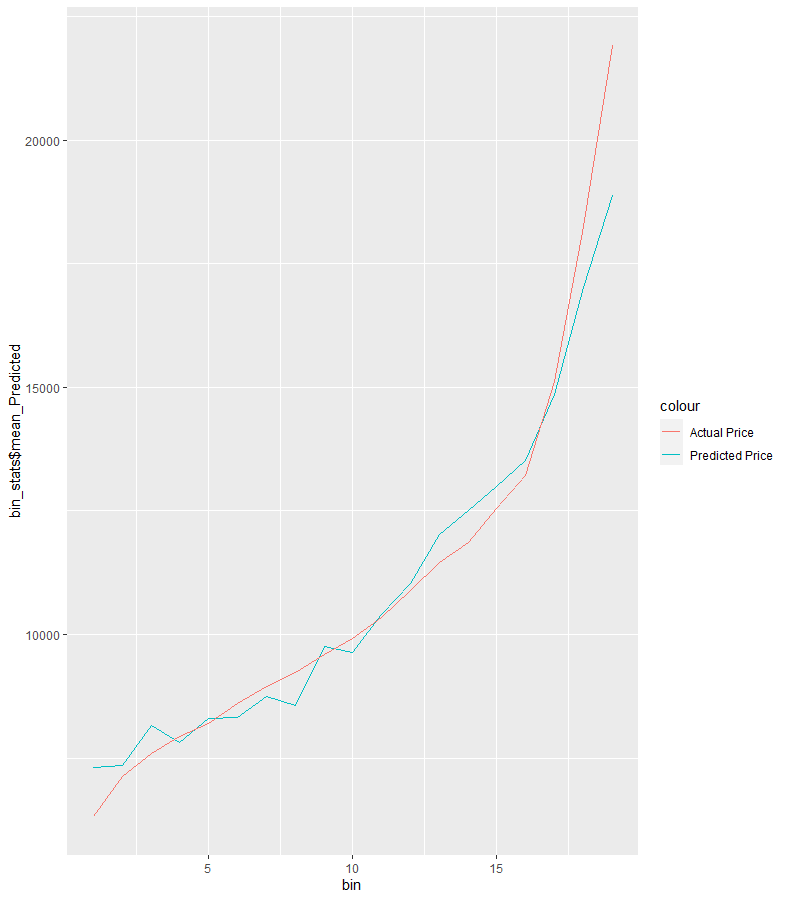
**Price = 12983.028 - 138.189 \* Age\_08\_04 – 0.021 \* KM + 2999.849 \* Fuel\_TypeDiesel +569.554 \* Fuel\_TypePetrol + 59.738 \* HP**

1. **What is the RMSE of the fitted model in the validation partition? Interpret the results.**

The RMSE of the fitted model in the validation partition is 1567.



The Root Mean Square Error represents deviation of the distance of the predicted values from the line of best fit. As the original data points are accumulated mostly around the best fit line, a higher RMSE value can also be the predicted value being away from the original values. RMSE indicates the deviation of the residuals from the line of best thus if the deviation is less, it depicts a good model has been generated to predict the values with good accuracy. While a model with high RMSE states the model may not be the best fit and has lower accuracy.



This graph shows the actual and predicted prices in the model. It can be seen that the predicted price line varies and has a lot of kinks from the actual price line.

1. **Using your fitted model, what is the predicted price of a diesel car 27 months old with 45,008 km driven, and a 90 HP engine? You may use Excel to assist you.**

Price = 12983.028 - 138.189 \* Age\_08\_04 – 0.021 \* KM + 2999.849 \* Fuel\_TypeDiesel +569.554 \* Fuel\_TypePetrol + 59.738 \* HP

Price = 12983.028 - 138.189 \* 27 – 0.021 \* 45008 + 2999.849 \* 1 +569.554 \* 0 + 59.738 \* 90

Price = 12983.028 - 3,731.103‬ - 945.168 + 2999.849 + 5,376.42‬

**Price = 16683.026$**

1. **Paste a copy of the R code below**

#load required libraries

library(forecast)

library(leaps)

library(ggplot2)

#load dataset

toyota.df <- read.csv("ToyotaCorolla-Exam1.csv")

View(toyota.df)

#Select columns for regression

toyota.df <- toyota.df[, c(3,4,5,6,7)]

#View(toyota.df)

#split into train and validation sets

set.seed(1)

nrow1 <- nrow(toyota.df)

train.index <- sample(nrow1, nrow1\*0.6)

#training dataset

train.df <- toyota.df[train.index,]

View(train.df)

#validation dataset

validation.df <- toyota.df[-train.index,]

View(validation.df)

#Run multiple Regression

toyota.lm <- lm(formula = Price ~ ., data = train.df)

options(scipen = 999)

summary(toyota.lm)

#stepwise

toyota.step <- step(toyota.lm,direction="forward")

summary(toyota.step)

#Using regression Subsets

models <- regsubsets(Price ~ ., data = train.df, nvmax = 16)

summary(models)

#Predict Price for validation set

toyota.lm.pred <- predict(toyota.lm, validation.df)

options(scipen=999, digits = 0)

#Display difference in predicted & actual values for 1st 10 rows

residuals <- validation.df$Price[1:10] - toyota.lm.pred[1:10]

data.frame("Predicted" = toyota.lm.pred[1:10], "Actual" = validation.df$Price[1:10],

"Residual" = residuals)

accuracy(toyota.lm.pred, validation.df$Price)

search <- regsubsets(Price ~ ., data = train.df, nbest=1,nvmax=dim(train.df)[2],method="exhaustive")

plot(search,scale="r2")

#calculate rmse by hand to show how it works. this will typically be done using accuracy()

residuals <- validation.df$Price - toyota.lm.pred

squaredResiduals <- residuals\*residuals

df <- data.frame("Predicted" = toyota.lm.pred, "Actual" = validation.df$Price,

"Residual" = residuals, "Squared Residuals" = residuals\*residuals)

rmse <- sqrt(mean(df$Squared.Residuals))

View(df)

#install required packages

install.packages("magrittr")

install.packages("Hmisc")

install.packages("dplyr")

install.packages("ggplot")

library(dplyr)

library(Hmisc)

library(ggplot)

#plot predicted price vs target price for range of prices

df <- df[order(-df$Actual),]

df$bin = as.numeric(cut2(df$Actual, g = 21))

table(df$bin)

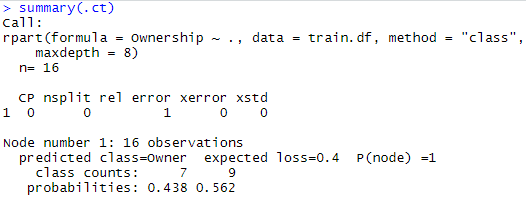
require(dplyr)

bin\_stats = df %>% group\_by(bin) %>% summarise(mean\_Actual = mean(Actual), mean\_Predicted = mean(Predicted), min\_Actual = min(Actual), min\_Predicted = min(Predicted), max\_Actual = max(Actual), max\_Predicted = max(Predicted) )

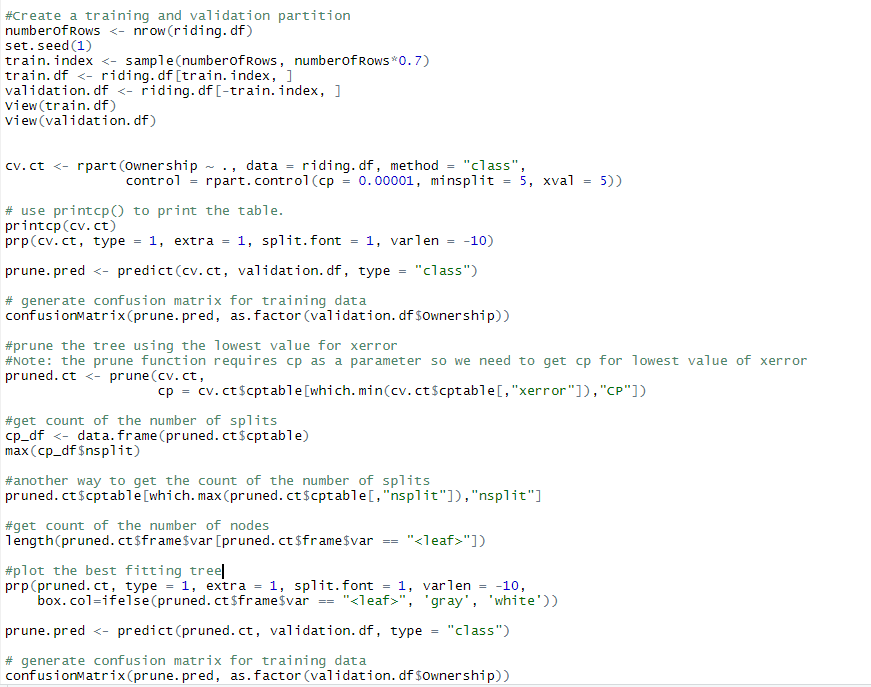
#Plotting actual vs predicted values for Training and Validation data

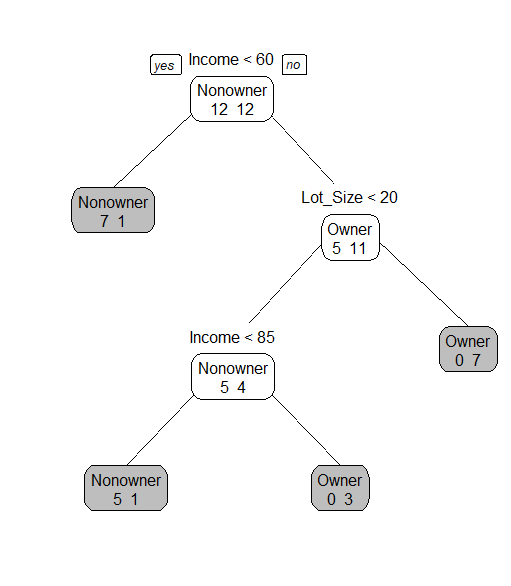
p1<- ggplot(bin\_stats, aes(bin)) + geom\_line(aes(y = bin\_stats$mean\_Predicted, color ="Predicted Price" )) + geom\_line(aes(y = bin\_stats$mean\_Actual, color = "Actual Price"))

p1



1. **Riding Mowers (20 points)**
2. **Use the RidingMower.csv dataset and create the best fitting decision tree with Ownership as the dependent target variable. Use 70% of the data for the training partition, 30% of the data for the validation partition, and a set.seed value of 1. Paste a copy of the best fitting decision tree below.**





This is the best fitting tree that can be built from the dataset with an accuracy of 100%

**Rules:**

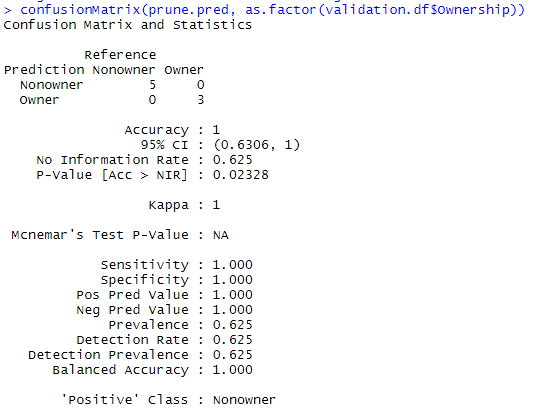
**Nonowner:**

* Income <60 = 1 => **Nonowner**
* Income <60 = 0 ; Lot\_Size <20 =1 => **Nonowner**
* Income <60 = 0 ; Lot\_Size <20 ; Income <85 =1 => **Nonowner**

**Owner:**

* Income <60 = 0 ; Lot\_Size <20 =0 => **Owner**
* Income <60 = 0 ; Lot\_Size <20 =1 ; Income <85 =0 => **Owner**

1. **Create the confusion matrix for the best fitting tree below. Label all parts of your confusion matrix.**



|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted** | |
|  |  | **Nonowner** | **Owner** |
| **Actual** | **Nonowner** | **5** | **0** |
| **Owner** | **0** | **3** |

The confusion matrix states that the model predicts accurately the Nonowners and owners for the validation set correctly.

TP +TN = 5+3=8

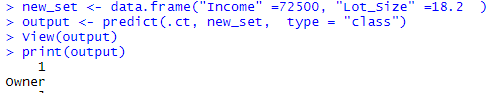
Total = 5+0+0+3

Accuracy = 8/8=1

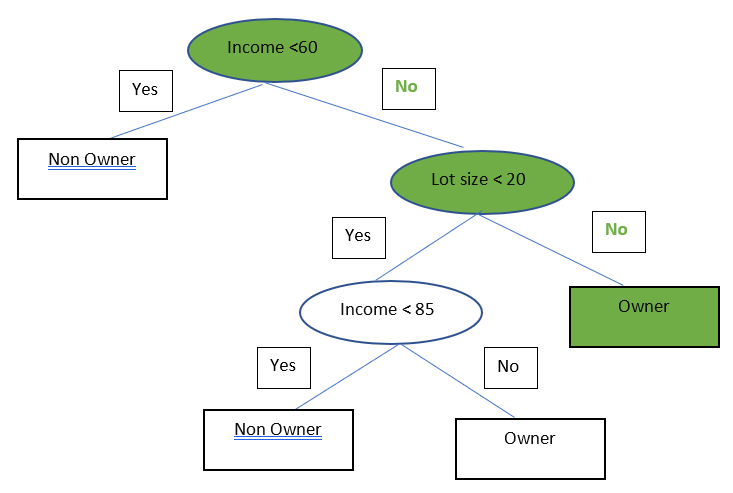
1. **What is the predictive accuracy of the decision tree?**

The predictive accuracy of the decision tree is 100%

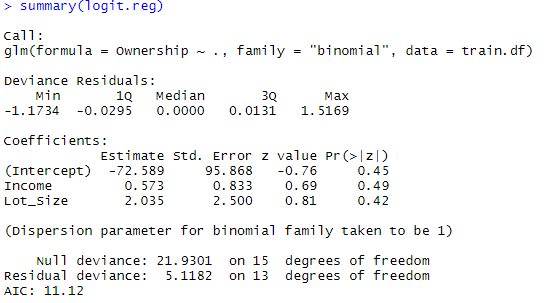
1. **What is the predicted outcome for someone with an income of $72,500 and a lot size of 18.2?**



The predicted outcome is that a person with income $72500 and lot size of 18.2 will be an **OWNER** of a riding mower.



1. **Use logistic regression to fit a model to predict ownership for Riding Mowers. Use the same rules as listed in part a to create training and validation partitions. What is the equation of the fitted model? Please include all parameter estimates in your equation.**

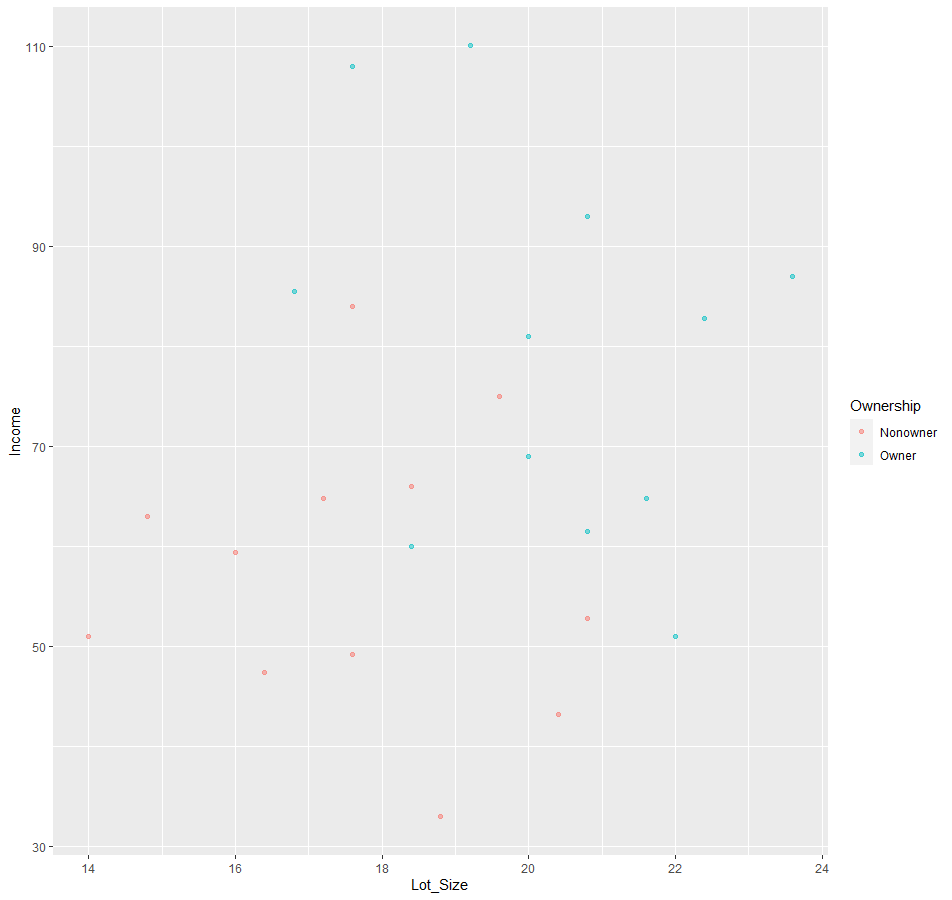




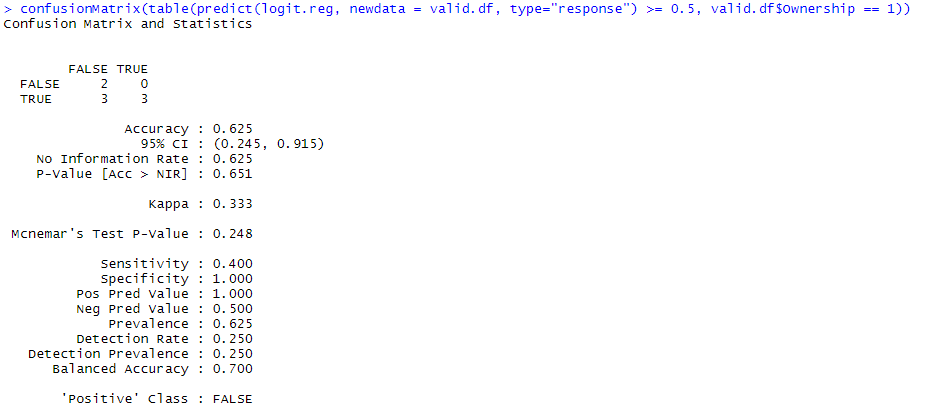
**ln(p/1-p) = -72.589 + 0.573 \* Income + 2.035 \* Lot\_Size**

where **p - probability of Ownership**

**(p/1-p) - odds ratio**



1. **Create the confusion matrix for the fitted logistic regression model below. Label all parts of your confusion matrix.**



|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted** | |
|  |  | **False** | **True** |
| **Actual** | **False** | **True Negative TN** | **False Positive FP** |
| **True** | **False Negative FN** | **True Positive TP** |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Predicted** | |
|  |  | **Nonowner** | **Owner** |
| **Actual** | **Nonowner** | **2** | **0** |
| **Owner** | **3** | **3** |

In the validation set, the model has predicted correctly that the 2 don’t own mowers and 3 own mowers. This is shown by the **True Negative**. However, it has predicted incorrectly that 3 owners as nonowners incorrectly. This is shown by the **False Negative**.

1. **What is the predictive accuracy of the logistic regression?**

The predictive accuracy of the decision tree is 62.5% with cutoff value of 0.5.

Accuracy = ( TN + TP ) / Total predicted

= ( 2 + 3 ) / 8 = 0.625

It has predicted 62% of the data correctly and 38% incorrectly.

1. **What is the predicted outcome for someone with an income of $72,500 and a lot size of 18.2? Show all of your work.**

**ln(p/1-p) = -72.589 + 0.573 \* Income + 2.035 \* Lot\_Size**

= -72.589 + 0.573 \* 72.500 + 2.035 \* 18.2

= -72.589 + 41.5425 + 37.037

ln(p/1-p) = 5.9905

(p/1-p) = e5.9905

(p/1-p) = 399.6144

p = 399.6144 - 399.6144p

400.6144p = 399.6144

p = 399.6144 / 400.6144

**p = 0.997**

Since the p value is greater than cut off value, with probability of 99.7% someone with $72,500 and lot size of 18.2 is an OWNER of a mower.

1. **Which model, decision tree or logistic regression, yields better predictive accuracy?**

The Decision tree has an accuracy of 100% and the logistic regression has accuracy of 62.5%. Decision tree should be considered a better model than logistic regression because of its accuracy, ease of implementation , understanding and decision tree bisects the space into smaller and smaller regions which results in a good model . On the other hand, logistic regression fits a single line which divides the space into two halves based on the cut off value . A single linear boundary can sometimes be a limiting factor of logistic regression.

1. **What, if anything, could be done to improve classification accuracy?**

* The presence of outliers in the train often reduces the accuracy of the model so removing the outlier is one of the ways to improve accuracy
* Boosting algorithm can be used to improve accuracy
* Penalty loss algorithm can be used where every wrong prediction made by the model will have a penalty associated with it
* Parameter tuning can be done by recursively changing the parameters such as cp , max depth , to find an optimum value which will improve the accuracy of the model