Business Data Mining

IDS 572, Spring 2020

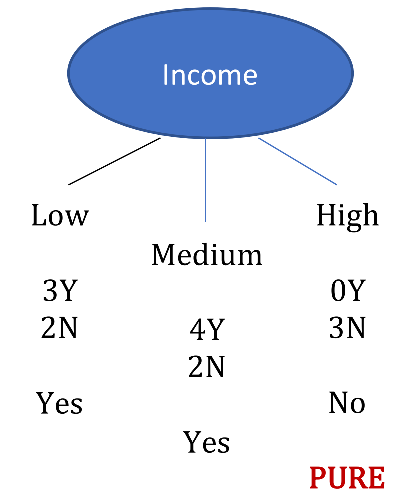
Team Members

Anuj Chanchlani (674153696)

Pratik Talreja (657488876)

Rashi Desai (663553314)

Q1 Create using the 1-rule, find the relevant sets of classification rules for the decision tree using the Gini index impurity measure



Error rate for Income(low) $= \frac{2}{5}$

Error rare for Income (medium) $= \frac{2}{6}$

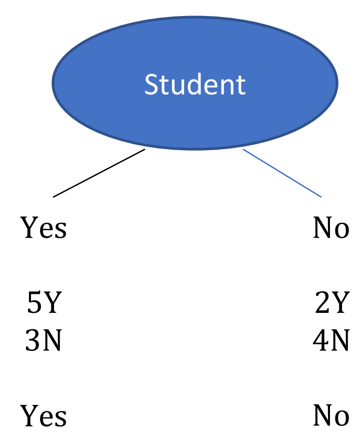
Total Error $= \frac{5}{14}\left(\frac{2}{5}\right) + \frac{6}{14}\left(\frac{2}{6}\right) + \frac{3}{14}(0) = \frac{1}{7} + \frac{1}{7} = \frac{2}{7} = 0.2857$

Gini (Low) = $1-((2/5)^2 + (3/5)^2 \qquad = 0.48$

Gini (Medium) = $1-((4/6)^2 + (2/6)^2 \qquad = 0.44$

Gini (High) = $1-((3/3)^2 + (0/3)^2 \qquad = 0$

Gini (Income) $= \frac{5}{14}(0.48) + \frac{6}{14}(0.44) + \frac{3}{14}(0) = 0.36$



Error rate for Student(Yes) $= \frac{3}{8}$

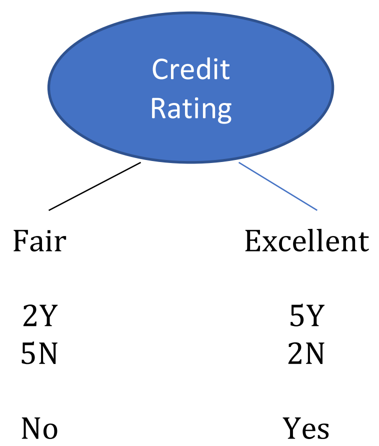
Error rate for Student (No) $= \frac{2}{6}$

Total Error $= \frac{8}{14}\left(\frac{3}{8}\right) + \frac{6}{14}\left(\frac{2}{6}\right) = \frac{3}{14} + \frac{2}{14} = \frac{5}{14} = 0.3571$

Gini (Yes) = 1-((3/8)^2 + (5/8)^2 = 0.46

Gini (No) = 1-((2/6)^2 + (4/6)^2 = 0.44

Gini (Student) $= \frac{8}{14}(0.46) + \frac{6}{14}(0.44) = 0.45$



Error rate for Credit Rating (Fair) $= \frac{2}{7}$

Error rare for Credit Rating (Excellent) $= \frac{2}{7}$

Total Error $= \frac{7}{14}\left(\frac{2}{7}\right) + \frac{7}{14}\left(\frac{2}{7}\right) = \frac{2}{7} = 0.2857$

Gini (Fair) = 1-((2/7)^2 + (5/7)^2 = 0.4081

Gini (Excellent) = 1-((5/7)^2 + (2/7)^2 = 0.4081

Gini (Credit Rating) $= \frac{7}{14}(0.4081) + \frac{7}{14}(0.4081) = 0.4081$

a) Out of the three sets of rules, the lowest misclassification rates belong to:

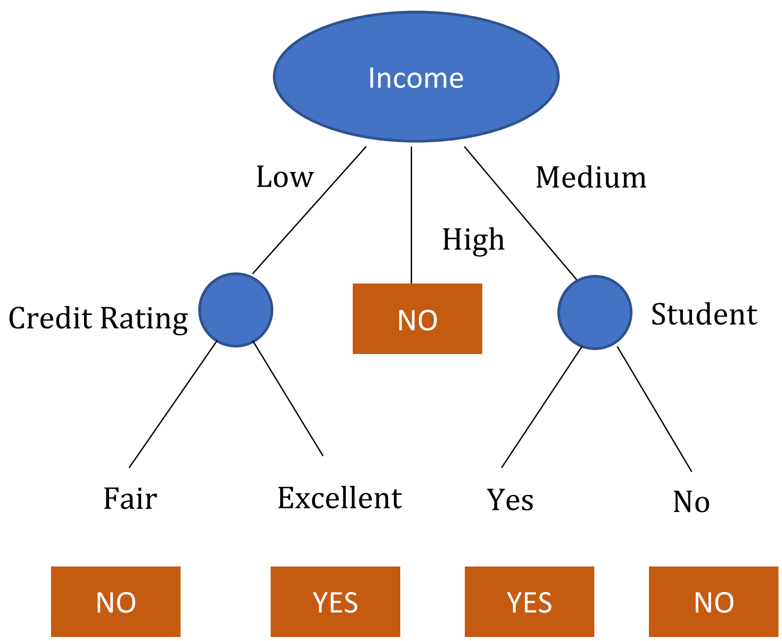
Income = 0.2857 and Credit Rating = 0.2857

b) Considering "buy-computer" as the target variable, we would select INCOME as the root in a decision tree that is constructed using the Gini index impurity measure as the Gini index for Income is the lowest equated to 0.36

c) Use the Gini index impurity measure and construct the full decision tree for this data set.

Income = Low -> Credit Rating

Income(Medium) -> Student



Gini (Fair) = 1-((2/2)^2 + (0/2)^2 = 0

Gini (Excellent) = 1-((2/3)^2 + (0/3)^2 = 0

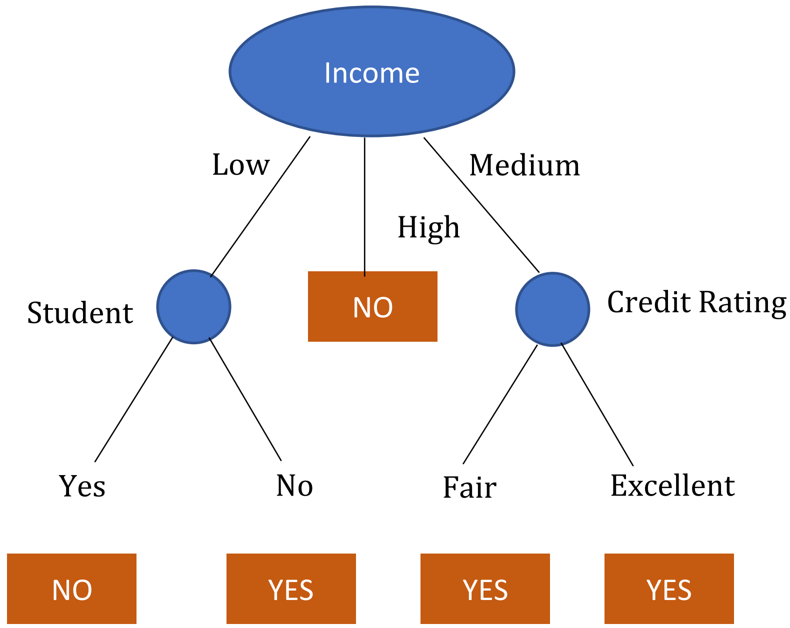
Gini (Credit Rating) = $\frac{2}{14}(0) + \frac{3}{14}(0)$ = 0

Gini (Yes) = 1-((4/4)^2 + (0/4)^2 = 0

Gini (No) = 1-((2/2)^2 + (0/2)^2 = 0

Gini (Student) = $\frac{4}{14}(0) + \frac{2}{14}(0)$ = 0

Income = Low -> Student ; Income = Medium -> Credit Rating



Gini (Yes) = $1-((2/3)^2 + (1/3)^2$ $= 0.44$

Gini (No) = $1-((2/2)^2 + (0/2)^2$ $= 0$

Gini (Student) = $\frac{3}{5}(0.44) + \frac{2}{5}(0)$ $= 0.26$

Gini (Fair) = $1-((2/3)^2 + (1/3)^2$ $= 0.44$

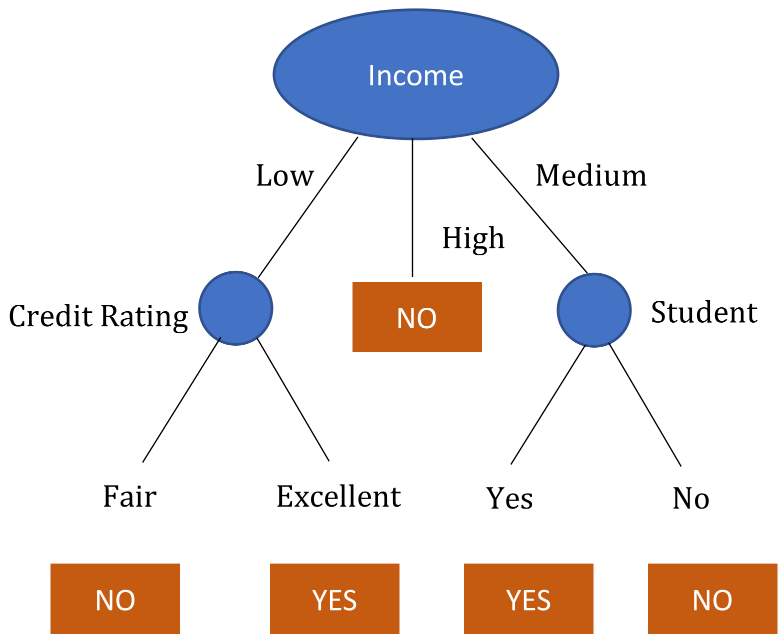
Gini (Excellent) = $1-((2/3)^2 + (0/3)^2$ $= 0.44$

Gini (Credit Rating) = $\frac{3}{6}(0.44) + \frac{3}{6}(0.44) = 0.44$

Based on the Gini index impurity measure,

credit rating split on income = low & student split on income =medium has a lower Gini index. Income(high) is a pure node.

The decision tree on the basis of Gini index impurity measure would be:



d) The decision rules would be:

1) If income = low & credit rating = fair, then buys computer = NO
2) If income = low & credit rating = excellent, then buys computer = YES
3) If income = medium & student = no, then buys computer = NO
4) If income = medium & student = yes, then buys computer = YES
5) If income = high, then buys computer = NO

Decision Rule	Support	Confidence
1	2/14 = 14.3%	2/7 = 28.57%
2	3/14 = 21.43%	3/7 = 42.85%
3	2/14 = 14.3%	2/7 = 28.57%
4	4/14 = 28.57%	4/7 = 57.14%
5	3/14 = 21.43%	3/7 = 42.85%

From the above table, we derive the highest support and confidence for Rule 4 and Rule 5

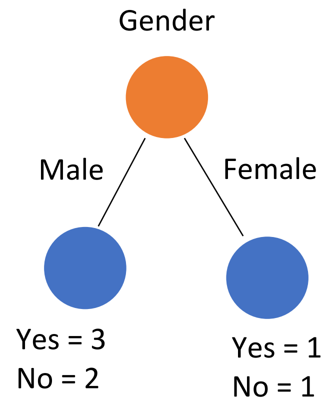
e) The accuracy of our decision tree model is 100% as we get all pure leaf nodes and we further do not split the tree. Also, the Gini index measure for this decision tree is lower as compared to the other tree.

Q2 Draw a decision tree learned by C5.0 for credit card approved

Target Variable: Credit card approved?

Info (T) = $-p_+\log_2 p_+ - p_-\log_2 p_-$

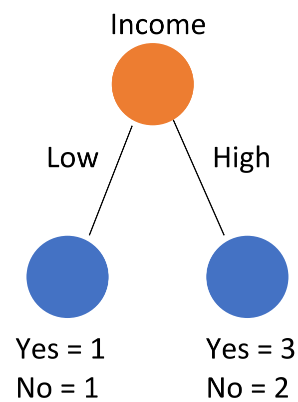
Info (T) = $-(4/7)\log_2(4/7) - (3/7)\log_2(3/7) = 0.9852$



$\text{Info}_{\text{Gender= Male}}(T) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$
$\text{Info}_{\text{Gender= Female}}(T) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$
$\text{Info}_{\text{Gender}}(T) = \frac{5}{7}(0.971) + \frac{2}{7}(1) = 0.9793$
Gain (Gender) = 0. 9852 - 0.9793 = 0.0059

Similarly, for income,

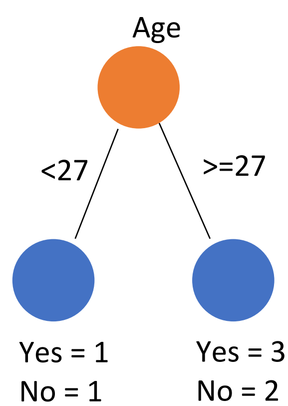
Info (T) = $-(4/7)\log_2(4/7) - (3/7)\log_2(3/7) = 0.9852$



$\text{Info}_{\text{Income=Low}}(T) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$
$\text{Info}_{\text{Income}} = \text{High}(T) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$
$\text{Info}_{\text{Income}}(T) = \frac{5}{7}(0.971) + \frac{2}{7}(1) = 0.9793$
Gain (Gender) = 0. 9852 - 0.9793 = 0.0059

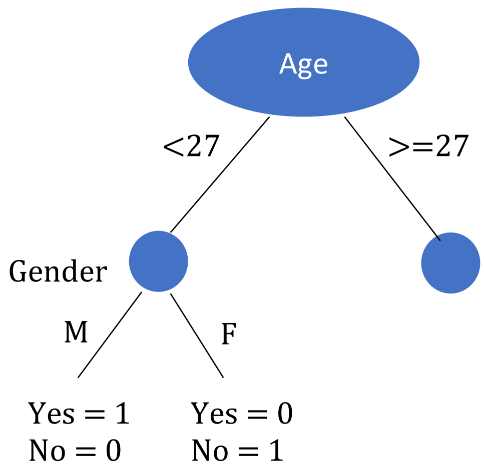
And for age,
Since age is a numeric attribute, we will place the split point halfway between values 22 and 38. If we take the average of 22-32 and 32-38 interval, we can consider 27 or 35 as our thresholds. We go with 27 here.

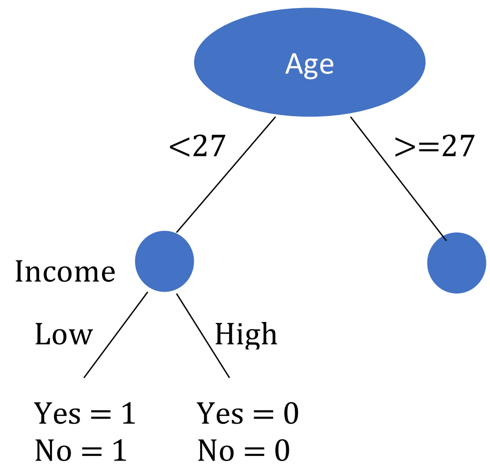
Info (T) = $-(4/7)\log_2(4/7) - (3/7)\log_2(3/7) = 0.9852$



$\text{Info}_{\text{Age<27}}(T) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$
$\text{Info}_{\text{Age>=27}}(T) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.971$
$\text{Info}_{\text{Gender}}(T) = \frac{5}{7}(0.971) + \frac{2}{7}(1) = 0.9793$
Gain (Gender) = 0. 9852 - 0.9793 = 0.0059

Since, we have the same gain for all the three attributes, we can choose any one at random for the root node. Let us consider **AGE** as the root node for our decision tree.





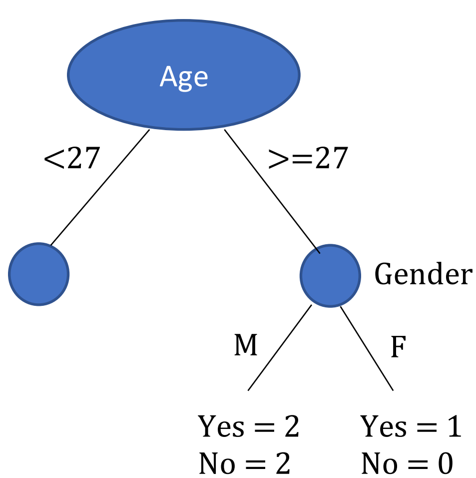
Split on Age <27 : Gender

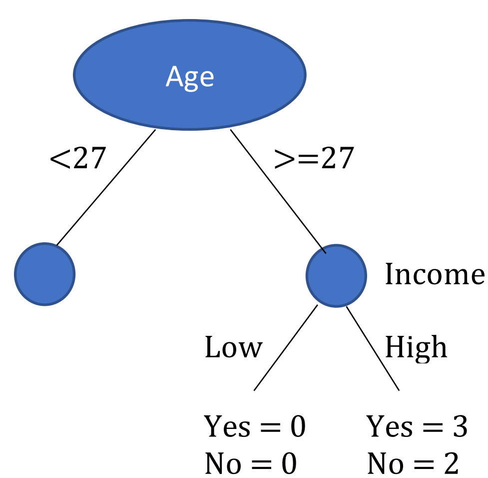
$\text{Info}(T) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) \quad = 1$
$\text{Info}(T1) = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1) \ = 0$
$\text{Info}(T2) = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1) \ = 0$
$\text{Info}_{\text{Gender}}(T) = \frac{1}{2}(0) + \frac{1}{2}(0) = 0$

$\text{Gain}(\text{Gender}) = 1 - 0 = 1$

Split on Age <27 : Income

$\text{Info}(T) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) \quad = 1$
$\text{Info}(T1) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$
$\text{Info}(T2) = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1) \ = 0$
$\text{Info}_{\text{Income}}(T) = \frac{2}{2}(1) + \frac{0}{2}(0) = 1$
$\text{Gain}(\text{Income}) = 1 - 1 = 0$





Split on Age >=27 : Gender

$\text{Info}(T) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.884$
$\text{Info}(T1) = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$
$\text{Info}(T2) = -(1/1)\log_2(1/1) - (0/1)\log_2(0/1) = 0$
$\text{Info}_{\text{Gender}}(T) = \frac{4}{5}(1) + \frac{1}{5}(0) = 0.8$

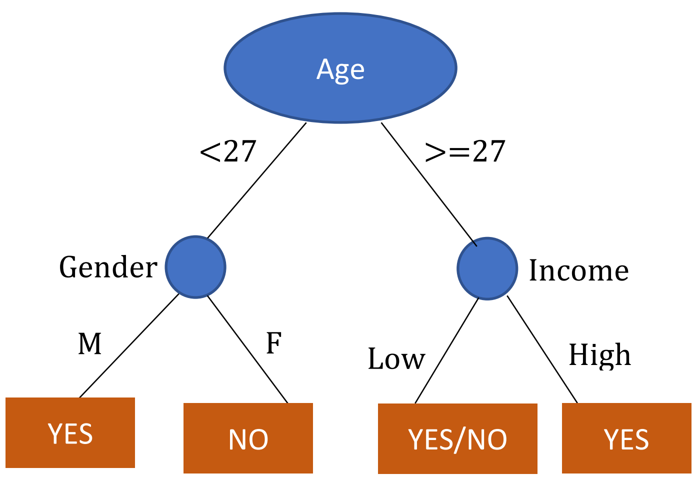
$\text{Gain(Gender)} = 0.884 - 0.8 = 0.084$

Split on Age >=27 : Income

$\text{Info}(T) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.884$
$\text{Info}(T1) = -(0/0)\log_2(0/0) - (0/0)\log_2(0/0) = 0$
$\text{Info}(T2) = -(3/5)\log_2(3/5) - (2/5)\log_2(2/5) = 0.884$
$\text{Info}_{\text{Income}}(T) = \frac{0}{5}(0) + \frac{5}{5}(0.884) = 0.884$
$\text{Gain(Income)} = 0.884 - 0.884 = 0$

From the calculations, we have the gain for split at age<27 for gender more than that at age>=27, therefore, the decision tree per C5.0 will run with gender at age < 27. Also, since the leaf nodes are pure for Gender after split, we stop further splitting.

The decision tree can be constructed as:



Q3 Perform operations on a College data set for 777 different universities and colleges in the US

library(dplyr) # data aggregates
library(gplots) # plot means with CI

#a
College = read.csv("College.csv")
dummy = read.csv("College.csv")
str(College)
#There are a total of 19 variables and 777 observations
#'data.frame':777 obs. of 19 variables:

#b
rownames (College) -> College [,1]
College = subset(College, select = -c(X))

#c
summary(College$Apps)

#Code Output

#Min. 1st Qu. Median Mean 3rd Qu. Max.
#81 776 1558 3002 3624 48094
#Mean(3002) is greater than median(1558).
Therefore, the data is right skewed. The data range is from 81 to 48094

summary(College$Accept)

#code Output

Min. 1st Qu. Median Mean 3rd Qu. Max.
72 604 1110 2019 2424 26330
#Mean(2019) is greater than median(1110)
#Therefore, the data is right skewed. Range is from 71 to 26330

summary(College$Enroll)
#Code Output

#Min. 1st Qu. Median Mean 3rd Qu. Max.
#35 242 434 780 902 6392
#Mean(780) is greater than median(434)

#Therefore, the data is right skewed. Range is from 35 to 6392

summary(College$Top10perc)

#code ouput

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#1.00   15.00   23.00   27.56   35.00   96.00
#Mean(27.56) is close to median(23). It is not completely normally distributed; The data is skewed to right

summary(College$Top25perc)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#9.0   41.0   54.0   55.8   69.0   100.0
#Median(54) is close to mean(55.8). It is close to being normally distributed. Range is 9 to 100

summary(College$F.Undergrad)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#139   992   1707   3700   4005   31643
#Mean (3700) is greater than median(1707). It is right skewed. The data range is from 39 to 31643

summary(College$P.Undergrad)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#1.0   95.0   353.0   855.3   967.0 21836.0
#Mean(855)is greater than median(353)
##Therefore, the data is right skewed. Data range is 1 to 21836

summary(College$Outstate)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#2340   7320   9990   10441   12925   21700
#Mean is 10441 > median is 9990.
#Therefore, the data is right skewed. Range is from 2340 to 21700.

summary(College$Room.Board)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#1780   3597   4200   4358   5050   8124
#Mean(4358) is close to median(4200). It is close to being normally distributed. Range is 1780 to 8124

summary(College$Books)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#96.0   470.0   500.0   549.4   600.0 2340.0
#Mean(549) is greater than median(500). It is not normally distributed. Range is from 96 to 2340

summary(College$Personal)

#code output

#Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#250   850   1200   1341   1700   6800
#Mean(1341) is greater than median(1200)
#Therefore, the data is right skewed. Range is from 250 to 6800

summary(College$PhD)
#code output
Min. 1st Qu. Median   Mean 3rd Qu.   Max.
#8.00   62.00   75.00   72.66   85.00 103.00
#Median(75) is greater than mean(72.66)
#Therefore, the data is left skewed. Data range is 8 to 103

summary(College$Terminal)

#code output
Min. 1st Qu. Median Mean 3rd Qu. Max.
#24.0 71.0 82.0 79.7 92.0 100.0
#Median (82) is greater than mean(79.7)
#Therefore, the data is left skewed. Data range is 24 to 100

summary(College$S.F.Ratio)
#code output
#Min. 1st Qu. Median Mean 3rd Qu. Max.
#2.50 11.50 13.60 14.09 16.50 39.80
#Mean(14.09) is close to median(13.6). it is close to being normally distributed. Data range is 2.5 to 39.8

summary(College$perc.alumni)
#code output
Min. 1st Qu. Median Mean 3rd Qu. Max.
#0.00 13.00 21.00 22.74 31.00 64.00
#Mean is 22.74 is close to Median of 21. Range is 0 to 64

summary(College$Expend)
#code output
#Min. 1st Qu. Median Mean 3rd Qu. Max.
#3186 6751 8377 9660 10830 56233
#Mean(9660) is greater than median(8377)
#Therefore, the data is right skewed. Data range is 3186 to 56233

summary(College$Grad.Rate)

#code output

#Min. 1st Qu. Median Mean 3rd Qu. Max.
#10.00 53.00 65.00 65.46 78.00 118.00
#Mean(65.46) is close to median(65).It is close to being normally distributed. Data Range is 10 to 118

#d.
acceptRate<-(College$Accept/College$Apps)
College <- cbind(College, acceptRate)
dummy <- cbind(dummy, acceptRate)

#e.
#By most selective we mean selecting universities with the lowest acceptance rate.

College1 <- dummy[order(acceptRate),]

top5 <- top_n(College1,-5)

	X	acceptRate
1	Princeton University	0.1544863
2	Harvard University	0.1561486
3	Yale University	0.2291453
4	Amherst College	0.2305904
5	Brown University	0.2573494

Now we print five most selective public institutions.

# Step 1: First subsetting only public institutions.
College2 <- dummy[which(College$Private == "No"),]

Step 2: Selecting five most selective

top5_1 <- top_n(College2,-5)

	X	acceptRate
1	Montclair State University	0.4076628
2	Rowan College of New Jersey	0.3746073
3	Stockton College of New Jersey	0.3928838
4	University of North Carolina at Chapel Hill	0.4100438
5	University of Virginia	0.3397060

#f.

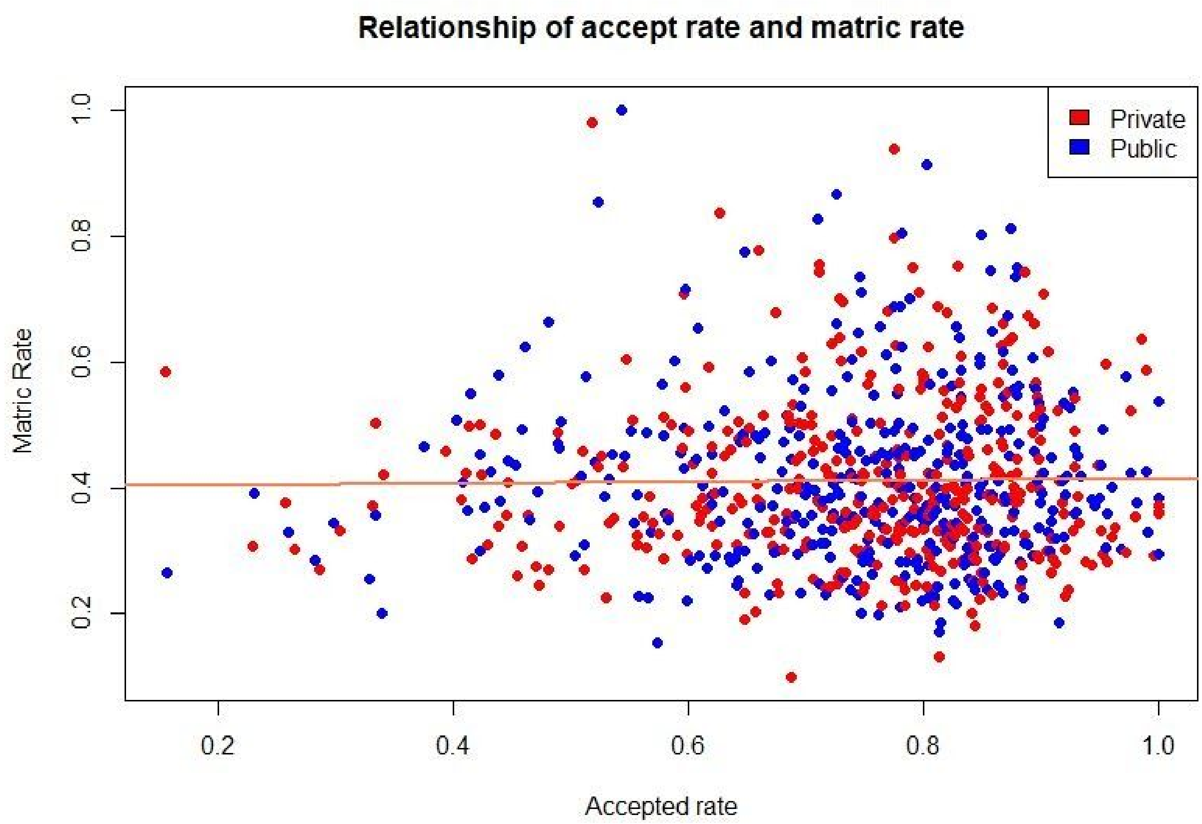
table(College1$Private)
mean(College1$acceptRate[College1$Private=="Yes"])#0.754
mean(College1$acceptRate[College1$Public=="Yes"])#0.726
#Public institutions are more selective on average. Since average acceptance rate of public universities is 72.6% and private institution is 75.4%

#G

matricRate<-College$Enroll/College$Accept
College <- cbind(College1, matricRate)

#H

```
plot(College$matricRate ~ College$acceptRate, col = c("red", "blue"),
    main="Relationship of accept rate and matric rate",
    xlab="Accepted rate",
    ylab=" Matric Rate",
    pch=16)
abline(lm(College$matricRate ~ College$acceptRate), col="coral", lwd=2.5)
lines(lowess(College1$Grad.Rate ~ College1$matricRate), col="green", lwd=2.5)
legend("topright", fill= c("red","blue"),
      legend = c("Private", "Public"),
      col = par("col"))
```



#I.

```
acc1 <- College$acceptRate[College$Private=="Yes"]
mat1 <- College$matricRate[College$Private=="Yes"]

acc2 <- College$acceptRate[College$Private=="No"]
mat2 <- College$matricRate[College$Private=="No"]

install.packages('corrplot')
library(corrplot)
```

cor.test(acc1,mat1)

#Code output
#Pearson's product-moment correlation

#data: acc1 and mat1
#t = 1.0235, df = 563, p-value = 0.3065
#alternative hypothesis: true correlation is not equal to 0
#95 percent confidence interval:
-0.03953148 0.12514064
#sample estimates:
 cor
#0.04309728

#correlation between acceptance rate and matriculation of private institutions is *weak, positive correlation.* cor=0.04
cor.test(acc2,mat2)

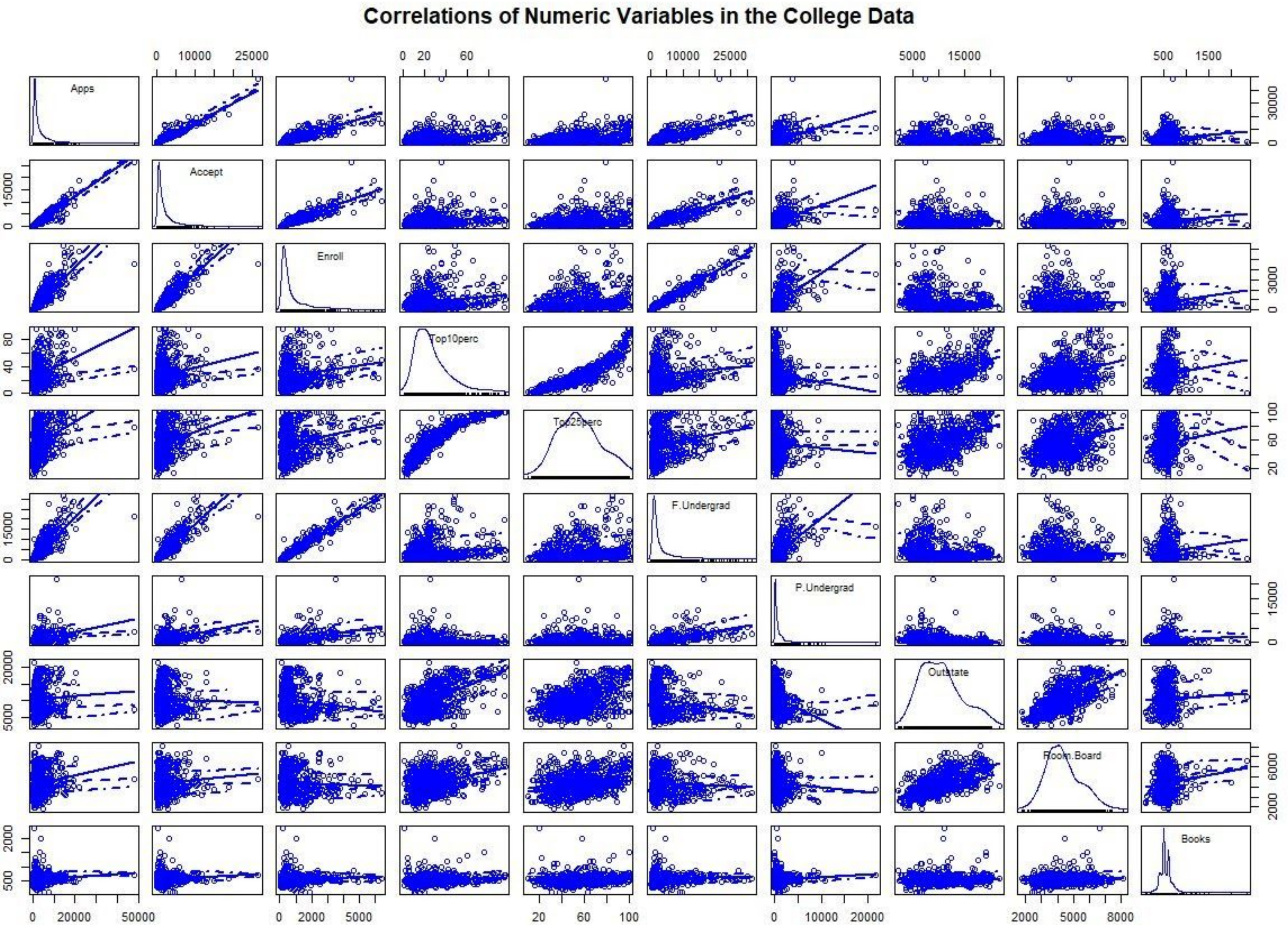
#Code Output
#Pearson's product-moment correlation

#data: acc2 and mat2
#t = -0.94132, df = 210, p-value = 0.3476
#alternative hypothesis: true correlation is not equal to 0
#95 percent confidence interval:
-0.19784190 0.07054418
#sample estimates:
cor
#-0.06482098

#The correlation between acceptance rate and matriculation of public institutions is *weak, negatively correlated*. cor= -0.064

#J
library(car) # advanced scatter plots

scatterplotMatrix(~Apps+Accept+Enroll+Top10perc+Top25perc+F.Undergrad+P.Undergrad+Outstate+Room.Board+Books, data=College1, main="Correlations of Numeric Variables in the College Data")

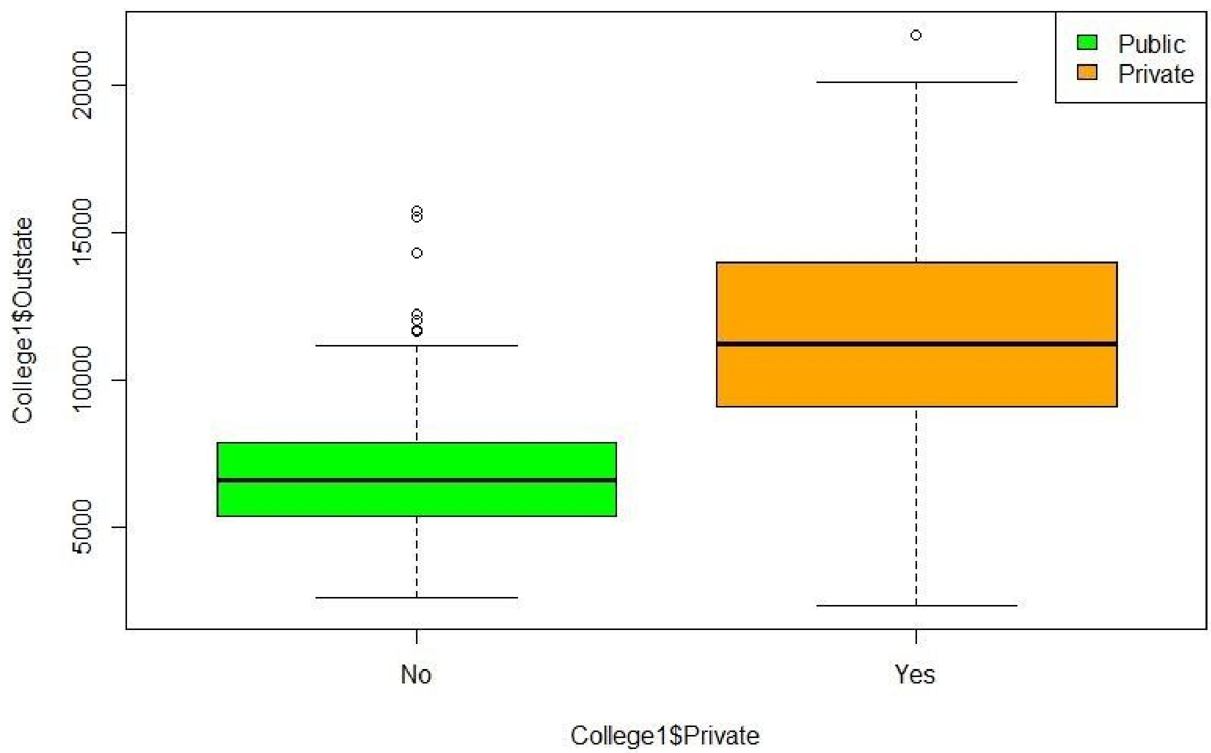


We move along each row from left to right, to find relationship between the two variables.
For example in case of Apps and Accept, the plot at position 1*2 represents the relations between the two.
If the plot shows an uphill pattern from left to right, this indicates a positive relation.
If the plot shows a downhill pattern from left to right, this indicates a negative relation.
if the plot doesn't show any kind of pattern then no relationship exists.

#K

boxplot(College1$Outstate~College1$Private, col = c("green", "orange"))#public institutions has few outliers above upper threshold, and private institutions
#has very few outliers above upper threshold
legend("topright", fill= c("green","orange"),

legend = c("Public", "Private"),
col = par("col"))



 # L

Elite<- rep ("No",nrow(College))

#Elite is the variable created. rep replicates the values in stated in first argument.(ie. No). the second argument is
#number of times, which is Number of rows in this case.

Elite[College$Top10perc > 50] <- "Yes"

#Above code is used for binning Top 10 perc in two categories ie. above and below 50%. High school exceeding 50% is categorized to Yes
#Below 50% is categorized to NO

Elite <- as.factor(Elite)

#as.factor is used to convert Elite variable fro character to factor data type. It is categorized in two levels(ie. Yes and No)

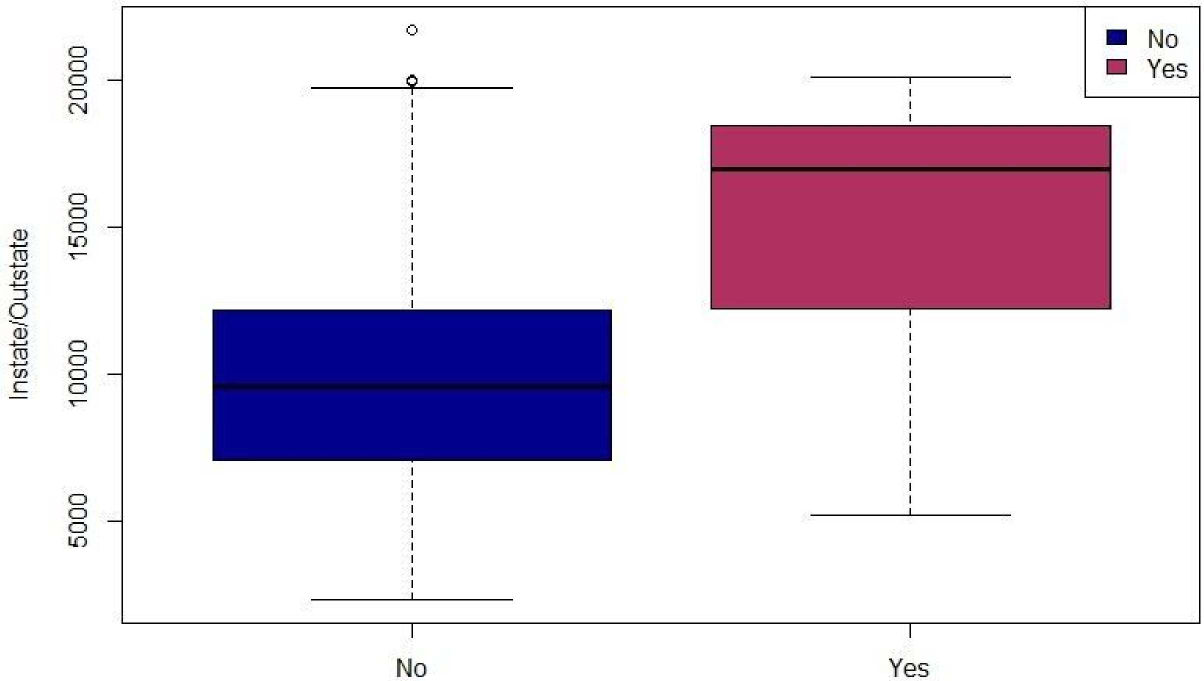
College1 <- data.frame(College1,Elite)

#its is used to combine variables and create a single data frame. In this case we join College dataset and Elite variable.

summary(College1$Elite)

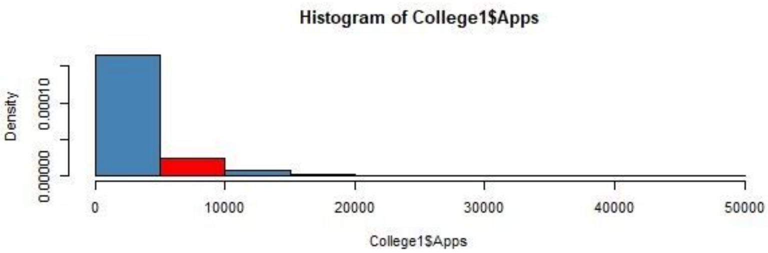
There are 78 Elite Universities.

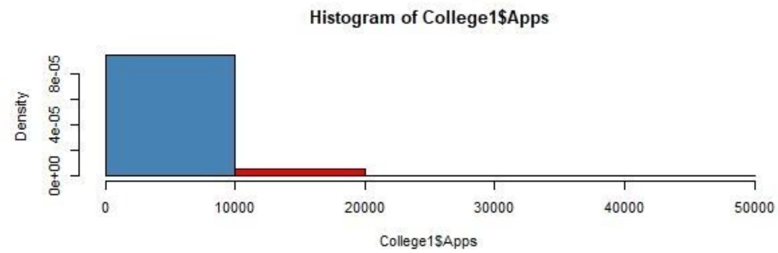
```
boxplot(College1$Outstate~College1$Elite, col = c("darkblue", "maroon"), xlab="Elite/Non Elite",ylab="Instate/Outstate" )
#Non-elite has few outliers above upper threshold, and elite universities has no outliers.
legend("topright", fill= c("darkblue","maroon"),
    legend = c("No", "Yes"),
    col = par("col"))
```

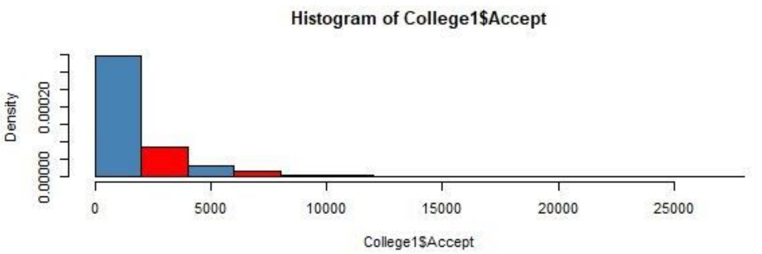


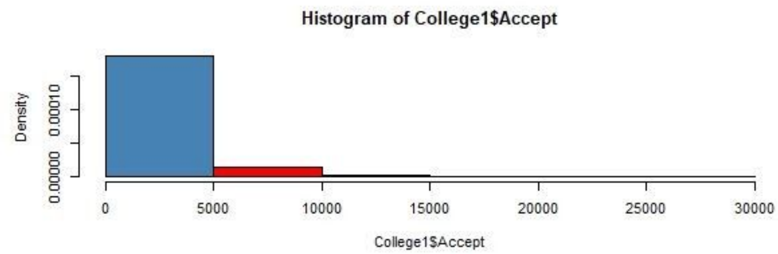
#M

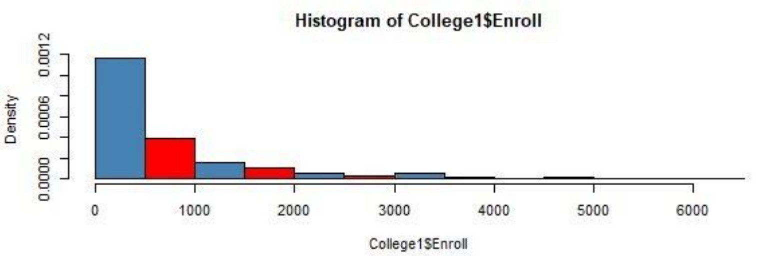
```
par(mfrow=c(4,2))

hist(College1$Apps, col=c("steelblue", "red"), freq=F)
hist(College1$Apps, col=c("steelblue", "red"), freq=F, breaks = 6)

hist(College1$Accept, col=c("steelblue", "red"), freq=F)
hist(College1$Accept, col=c("steelblue", "red"), freq=F, breaks = 6)

hist(College1$Enroll, col=c("steelblue", "red"), freq=F)
hist(College1$Enroll, col=c("steelblue", "red"), freq=F, breaks = 6)

hist(College1$PhD, col=c("steelblue", "red"), freq=F)
hist(College1$PhD, col=c("steelblue", "red"), freq=F, breaks = 6)
```

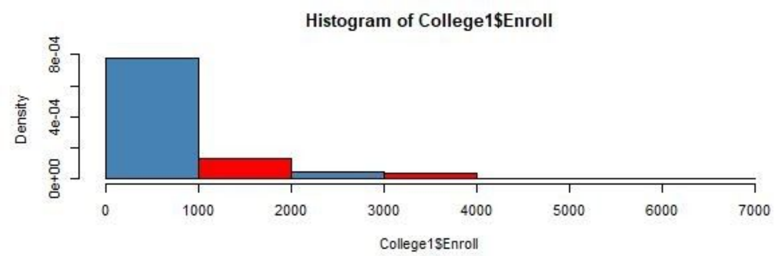

Histogram of College1$Apps
Density
0.00000 0.00010
0 10000 20000 30000 40000 50000
College1$Apps

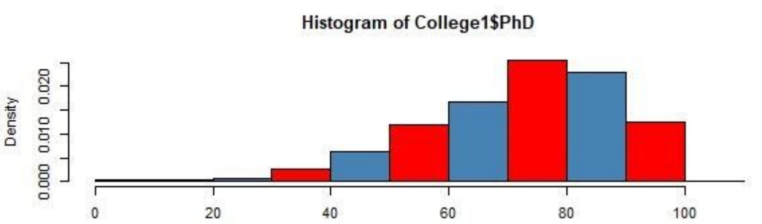

Histogram of College1$Apps
Density
0e+00 4e-05 8e-05
0 10000 20000 30000 40000 50000
College1$Apps

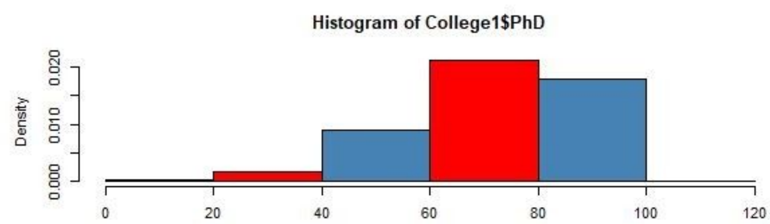

Histogram of College1$Accept
Density
0.00000 0.00020
0 5000 10000 15000 20000 25000
College1$Accept


Histogram of College1$Accept
Density
0.00000 0.00010
0 5000 10000 15000 20000 25000 30000
College1$Accept


Histogram of College1$Enroll
Density
0.0000 0.0006 0.0012
0 1000 2000 3000 4000 5000 6000
College1$Enroll


Histogram of College1$Enroll
Density
0e+00 4e-04 8e-04
0 1000 2000 3000 4000 5000 6000 7000
College1$Enroll


Histogram of College1$PhD
Density
0.000 0.010 0.020
0 20 40 60 80 100


Histogram of College1$PhD
Density
0.000 0.010 0.020
0 20 40 60 80 100 120

Q4 Perform analysis on Auto Dataset

```
#importing libraries
library(corrplot)
library(dplyr)
library(gmodels)
library(gplots)
library(psych)
library(corrplot) # for correlation plot
library(Hmisc)
library(ggplot2) # for plots
library(ggthemes)
install.packages("corrplot")
library(corrplot)
# Assigning dataframe to the data variable
data <- read.csv("Auto.csv")
```

#A

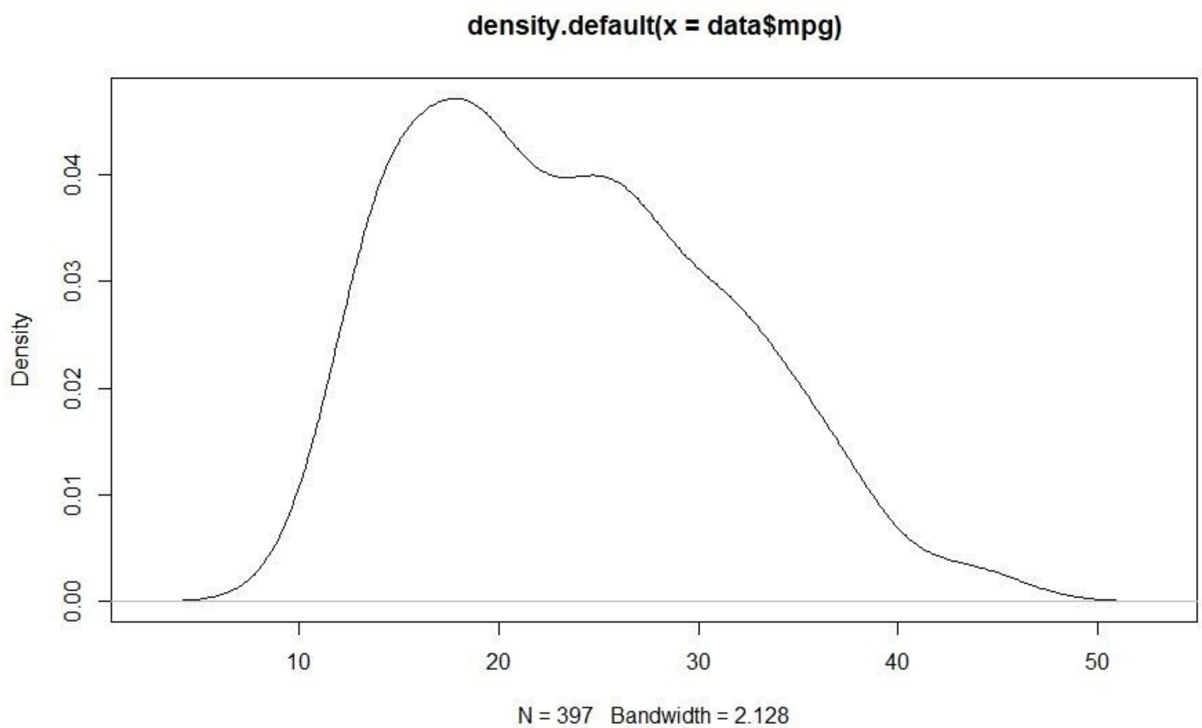
```
# Checking the data
View(data) # There are some cases where "?" are present in the data. We need to remove them

# we will first replace all "?" with "NA" and then remove all NA.
data[data == "?"] <- NA

# counting number of NA
sum(is.na(data)) # there are five instances of misisng values.

# removing all the missing values.
data <- na.omit(data)

View(data)
# dataframe is now free of all missing values.
```

#B

```
# for quantitative we can write
is.numeric(datasetname$variablename)
# for qualitative we can write
is.factor(datasetname$variablename)
```

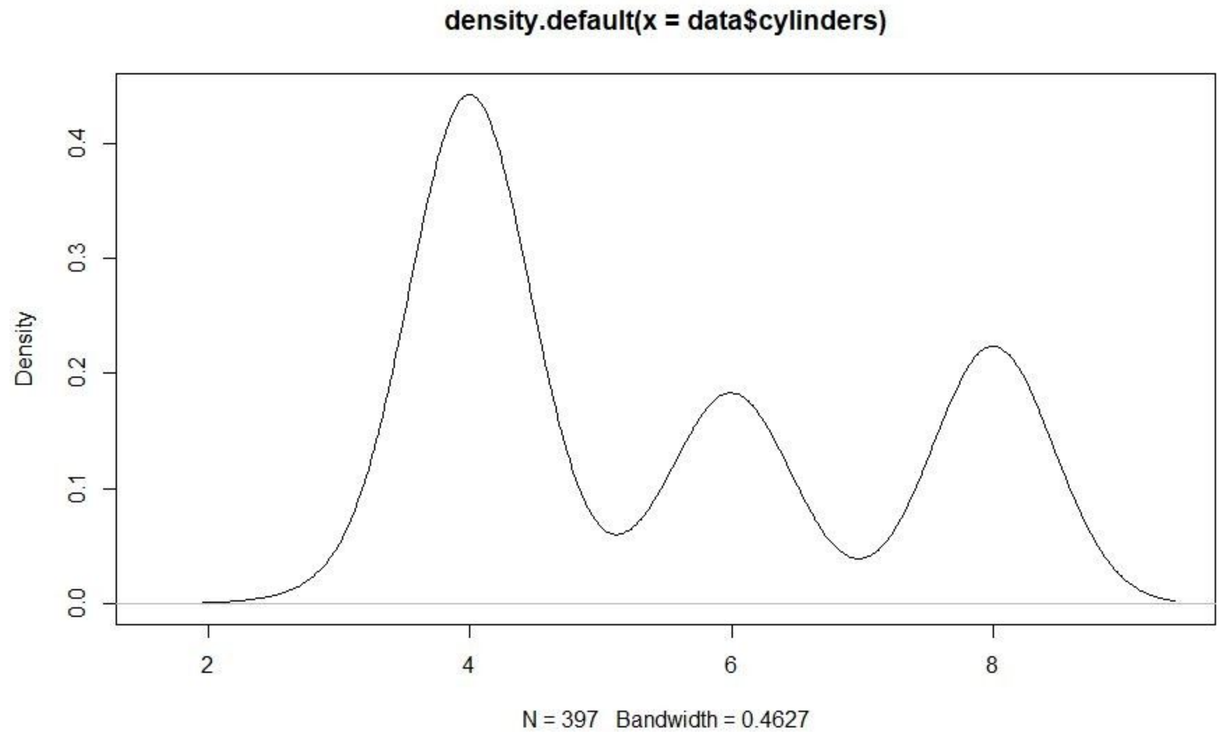
Initially, mpg,cylinders, displacement, weight,acceleration, year, origin are numeric
Horsepower and name are factors.
Now by doing analysis finally whether variables are quantitative or qualitative is decided.

mpg. How to check it?
psych::describe(data$mpg) # mean and median both are close to one another indicating near normal distribution.
Since mean and median are almost equal, it indicates near normal distribution.
Let's plot density plot.
plot(density(data$mpg)) # Proves our speculation.



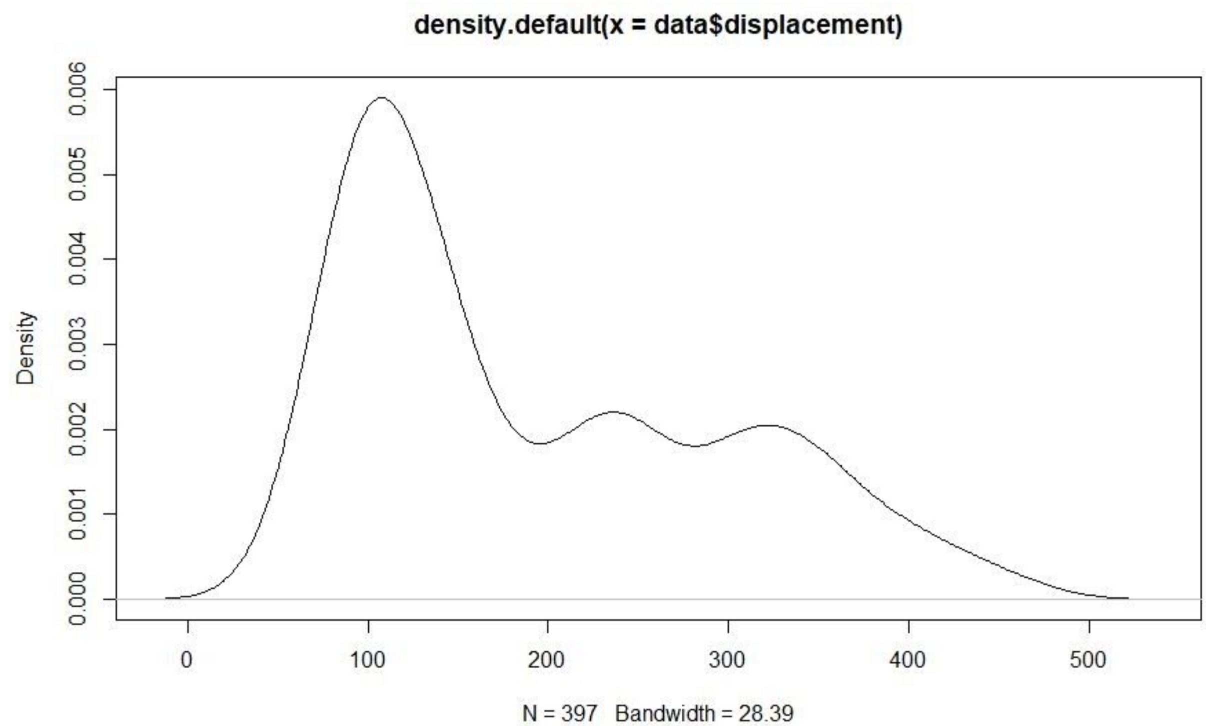
Also we can perform mathematical operations on the mpg values and it will still hold relevance.
hence mpg should be quantitative.

cylinders. How to check it?
psych::describe(data$cylinders)
Negative value of kurtosis indicates there might be some abnormality in the curve.
Let's plot density plot.
plot(density(data$cylinders)) # Proves our speculation.



# The plot has three peaks i.e. a tri-modal distribution
# It would be wise to check whether it can be qualitative or not.
dummy <- as.factor(data$cylinders)
# We are getting five defined levels.hence cylinders should not be quantitative. Rather it should be qualitative.
# Converting cylinders into qualitative
data$cylinders <- as.factor(data$cylinders)

displacement. How to check it?
psych::describe(data$displacement)
# Let's plot density plot.
plot(density(data$displacement))



range(data$displacement) # we cannot define displacement in less no of defined levels. Also when we perform mathematical
operations on the displacement values it will hold some relevance. Hence displacement should be quantitative.

# horsepower. How to check it?
table(data$horsepower)

```
  ? 100 102 103 105 107 108 110 112 113 115 116 120 122 125 129 130 132 133 135 137 138 139 140 142 145 148 149 150
  5  17   1   1  12   1   1  18   3   1   5   1   4   1   3   2   5   1   1   1   1   1   2   7   1   7   1   1  22
152 153 155 158 160 165 167 170 175 180 190 193 198 200 208 210 215 220 225 230  46  48  49  52  53  54  58  60  61
  1   2   2   1   2   4   1   5   5   5   3   1   2   1   1   1   3   1   3   1   2   3   1   4   2   1   2   5   1
 62  63  64  65  66  67  68  69  70  71  72  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
  2   3   1  10   1  12   6   3  12   5   6   3  14   4   1   6   2   7   2   1   4   6   9   5   2  19   1  20   1
 92  93  94  95  96  97  98
  6   1   1  14   3   9   2
```

It would be wise to convert it into quantitative since we cannot segregagte the horsepower
into less number of defined levels.
Hence it should be quantitative. Again performing mathematical operations on the horsepower will hold some relevance.

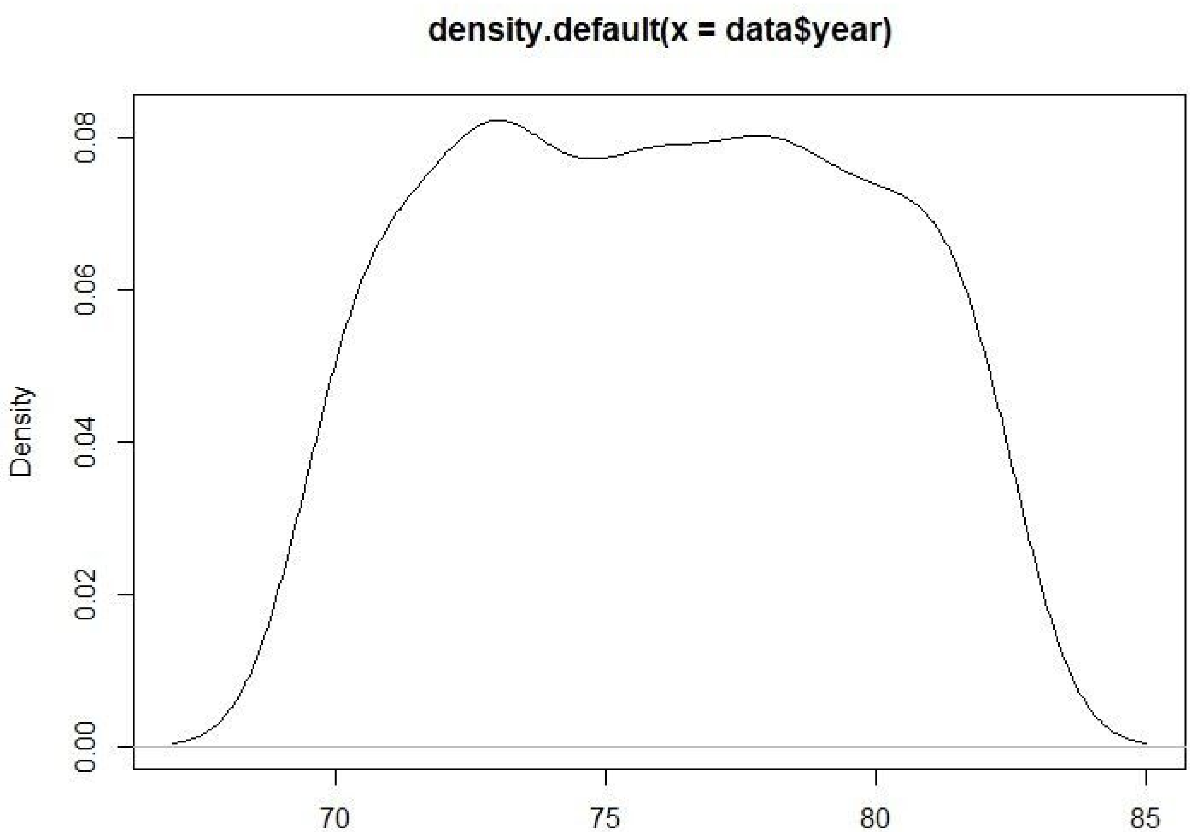
data$horsepower <- as.numeric(data$horsepower)

weight. How to check it?
psych::describe(data$weight)
summary(data$weight) # weight describes weight of cars, Different cars have different weights. Hence it should be numeric

we cannot categorize it into defined levels. Arithmethic operation on the weight will hold relevance.

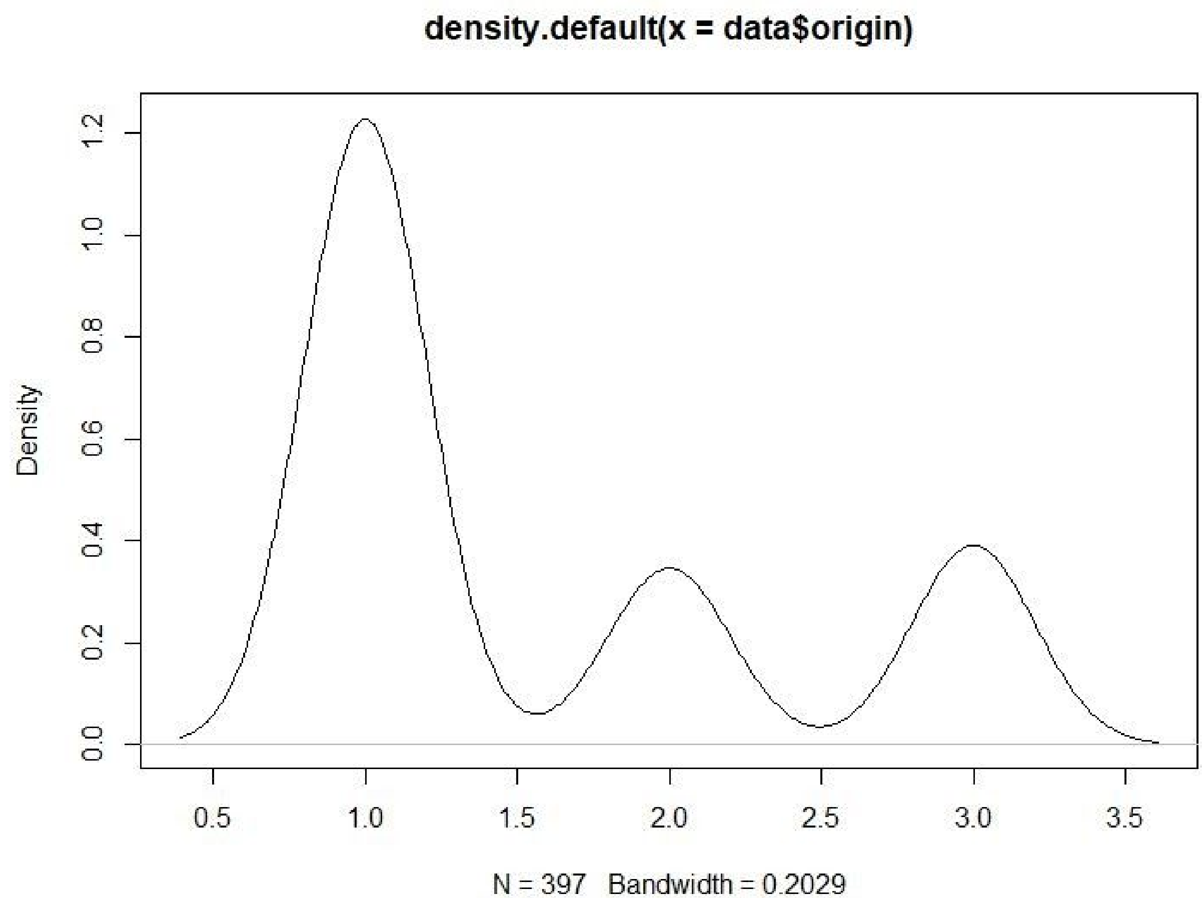
acceleration. How to check it?
psych::describe(data$acceleration)
summary(data$weight) # acceleration describes speed of cars, Different cars have different accelerations. Hence it should be numeric
we cannot categorize it into defined levels. Arithmethic operation on the acceleration will hold relevance.

summary(data$year)
plot(density(data$year))



Performing arithmetic operations on the year variable will hold no relevance. Hence, year should be qualitative.
converting year to qualitative
data$year <- as.factor(data$year)

summary(data$origin)
plot(density(data$origin))



we se an abnormal curve. Hence it would be wise to convert origin into factor.
Also performing artihtmetic operations on origin will hold no relevance.
converting origin to qualitative
data$origin <- as.factor(data$origin)

#C

psych::describe(data$mpg)
range for mpg is 37.6

psych::describe(data$displacement)
range for displacement is 387

psych::describe(data$horsepower)
range for horsepower is 92

psych::describe(data$weight)
range for weight is 3527

psych::describe(data$acceleration)
range for acceleration is 16.8

#D

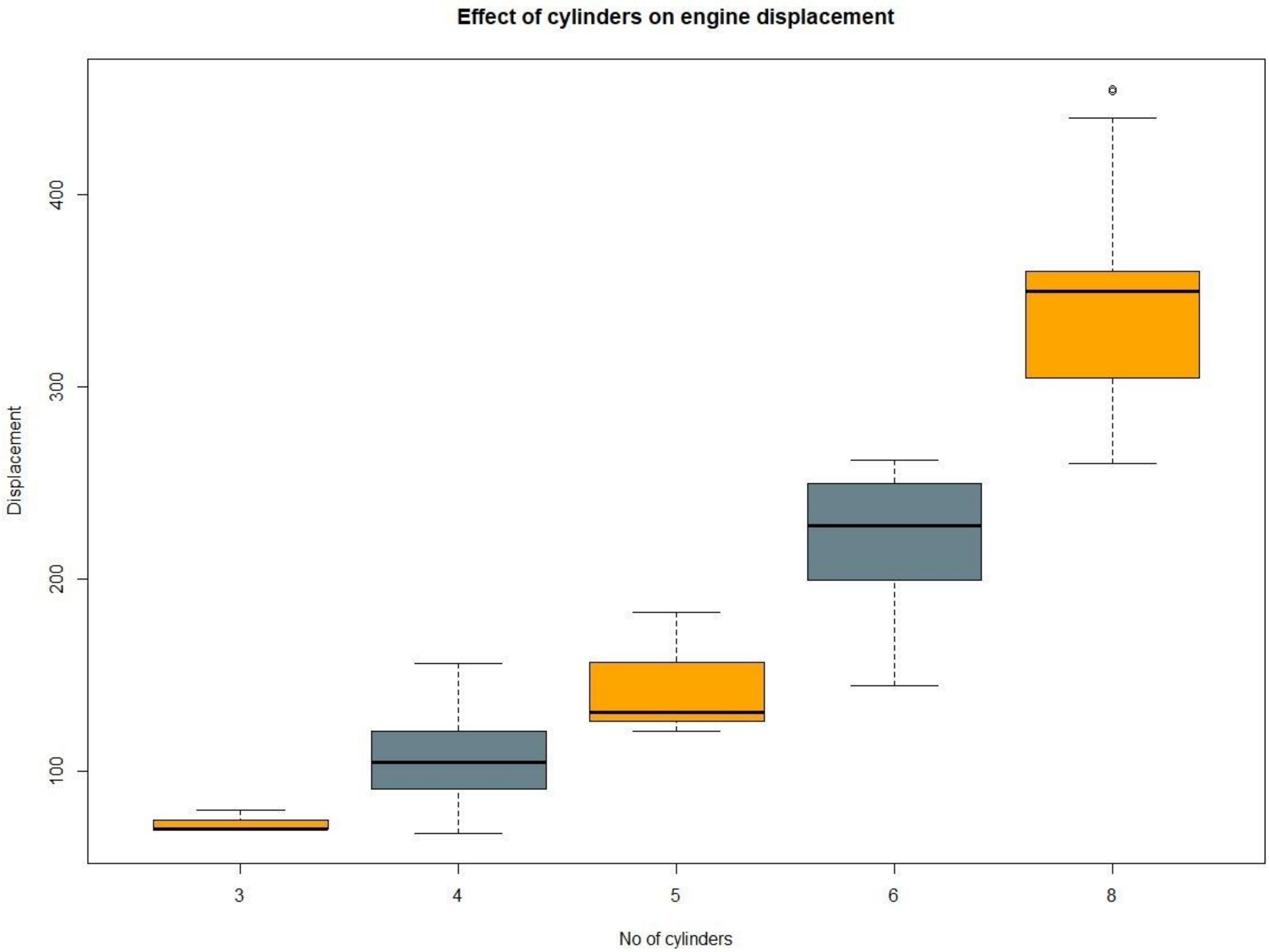
```
# mean of mpg is 23.45. standard deviation is 7.81
# mean of displacement is 194.41. standard deviation is 104.64
# mean of horsepower is 52.16. standard deviation is 29.5
# mean of weight is 2977.58. standard deviation is 849.4
# mean of acceleration is 15.54. standard deviation is 2.76
```

#E

```
newdata <- data[-c(10:84),]

psych::describe(newdata$mpg) # range is 35.6 , mean is 24.37, standard deviation is 7.88.

psych::describe(newdata$displacement) # range is 387 , mean is 187.75, standard deviation is 99.94.

psych::describe(newdata$horsepower) # range is 92 , mean is 51.63, standard deviation is 29.73.

psych::describe(newdata$weight) # range is 3348 , mean is 2939.64, standard deviation is 812.65.

psych::describe(newdata$acceleration) # range is 16.3 , mean is 15.72, standard deviation is 2.69.
```

#f draw plots.

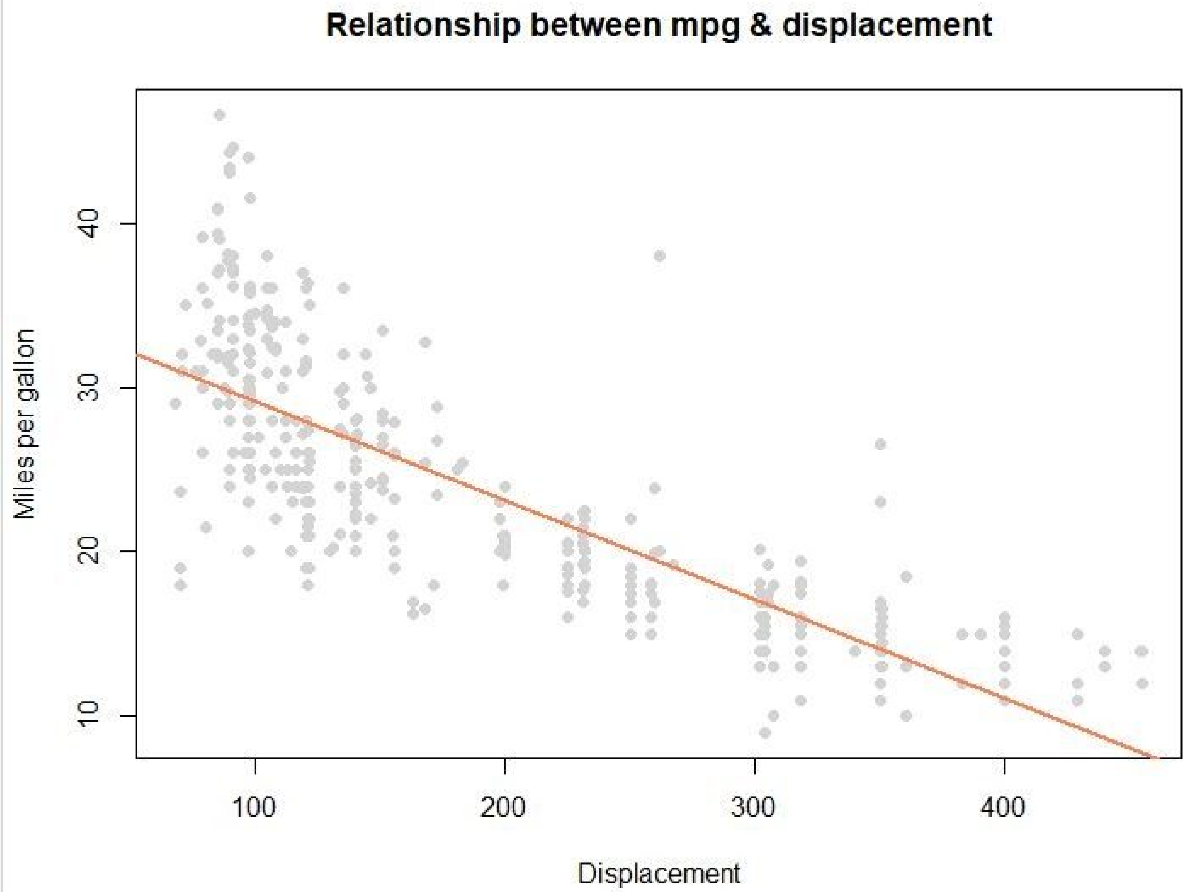
```
# lets analyze relation between no of cylinders and displacement

boxplot(data$displacement ~ data$cylinders, data=data, main="Effect of cylinders on engine displacement", xlab="No of cylinders", ylab="Displacement",col=c("orange", "lightblue4"))
```



from the boxplots we can say that more the no of cylinders, more is the displacement produced by it.

lets analyze relation between mpg and displacement

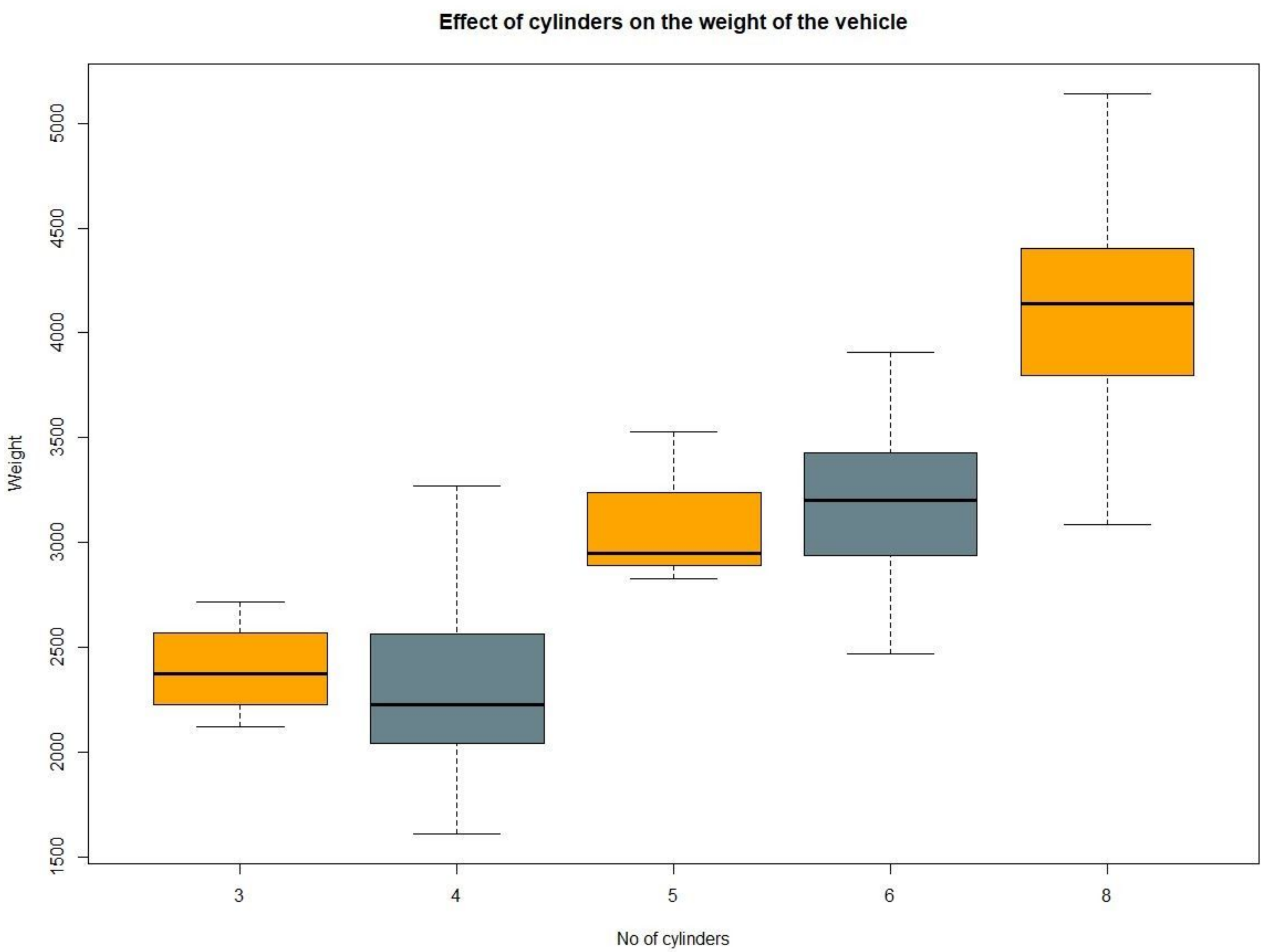
```
relation <- lm(mpg ~ displacement, data = data)
plot(data$mpg ~ data$displacement, col="lightgray", main="Relationship between mpg &
displacement", xlab="Displacement", ylab="Miles per gallon", pch=16)
abline(relation, col = "coral" , lwd = 2.5)
```



from the plot we see that vehicles which produce more engine displacement tend to have low fuel economy i.e mpg.

let's analyze relation between no of cylinders and weight

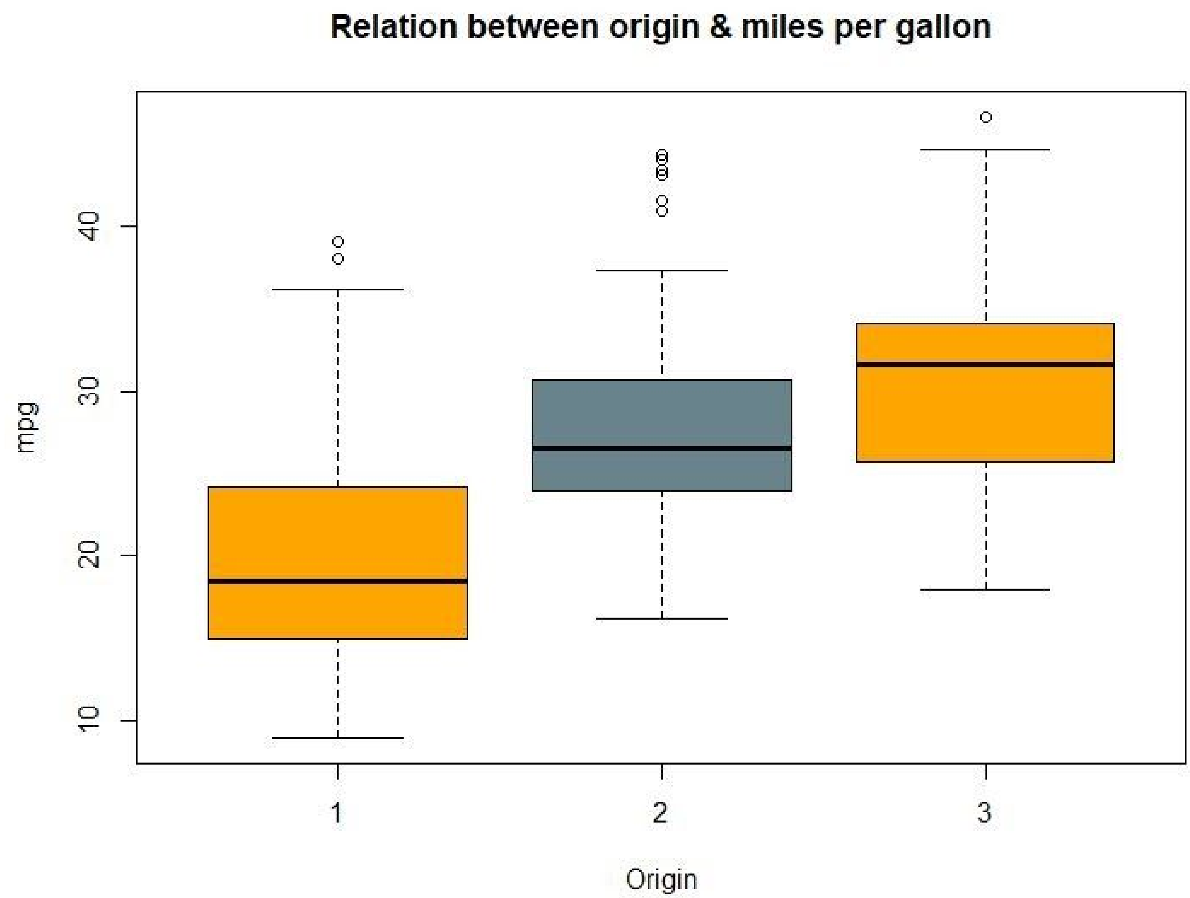
boxplot(data$weight ~ data$cylinders, data=data, main="Effect of cylinders on the weight of the vehicle", xlab="No of cylinders", ylab="Weight",col=c("orange", "lightblue4"))



from the boxplots we see that as no of cylinders increase the weight of the vehicle also increases.

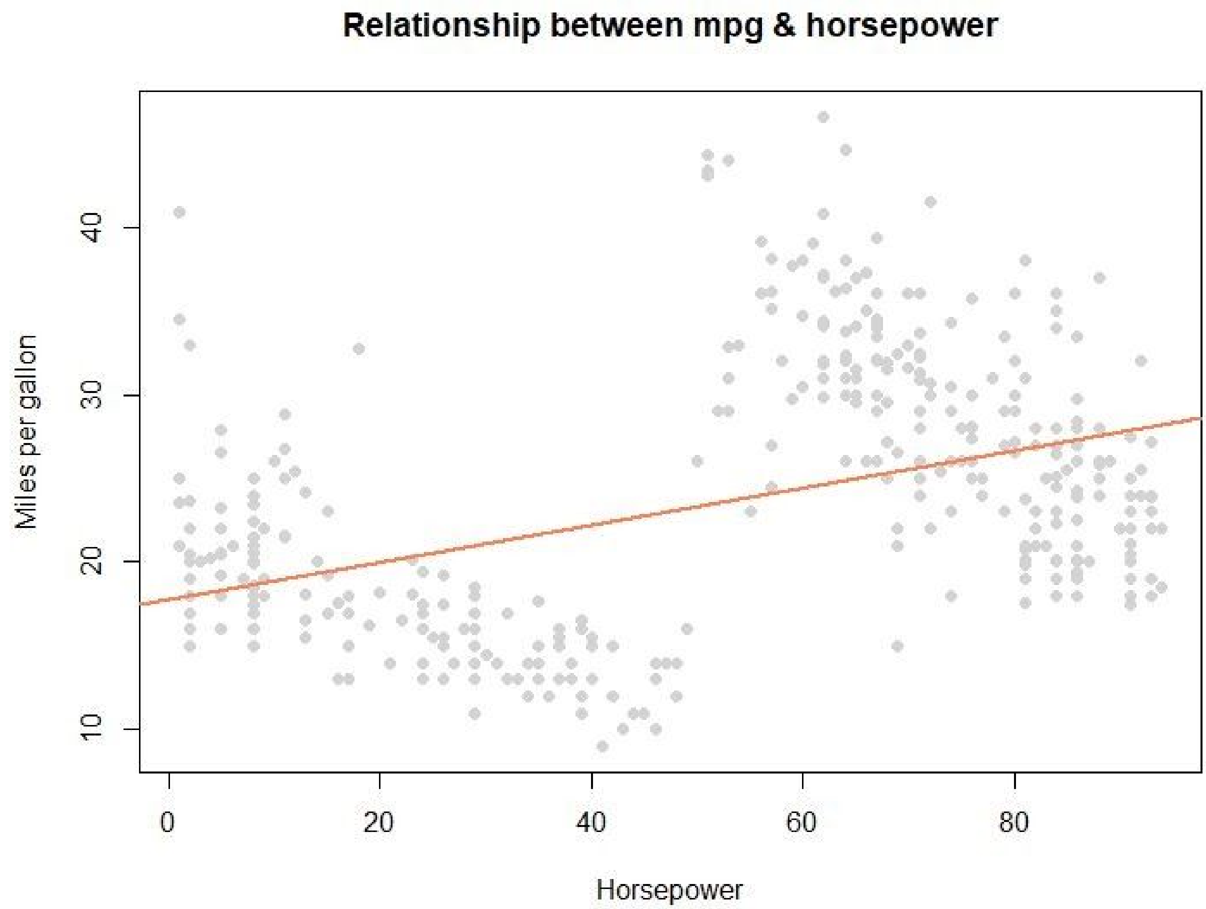
lets analyze relation between no of cylinders and miles per gallon

boxplot(data$mpg ~ data$origin, data=data, main="Relation between origin & miles per gallon", xlab="Origin", ylab="mpg",col=c("orange", "lightblue4"))
# From the boxplots we see that vehicles with origin 3 have mpg greater than those with origin 2 and origin 1.



lets analyze the relation between horsepower and miles per gallon

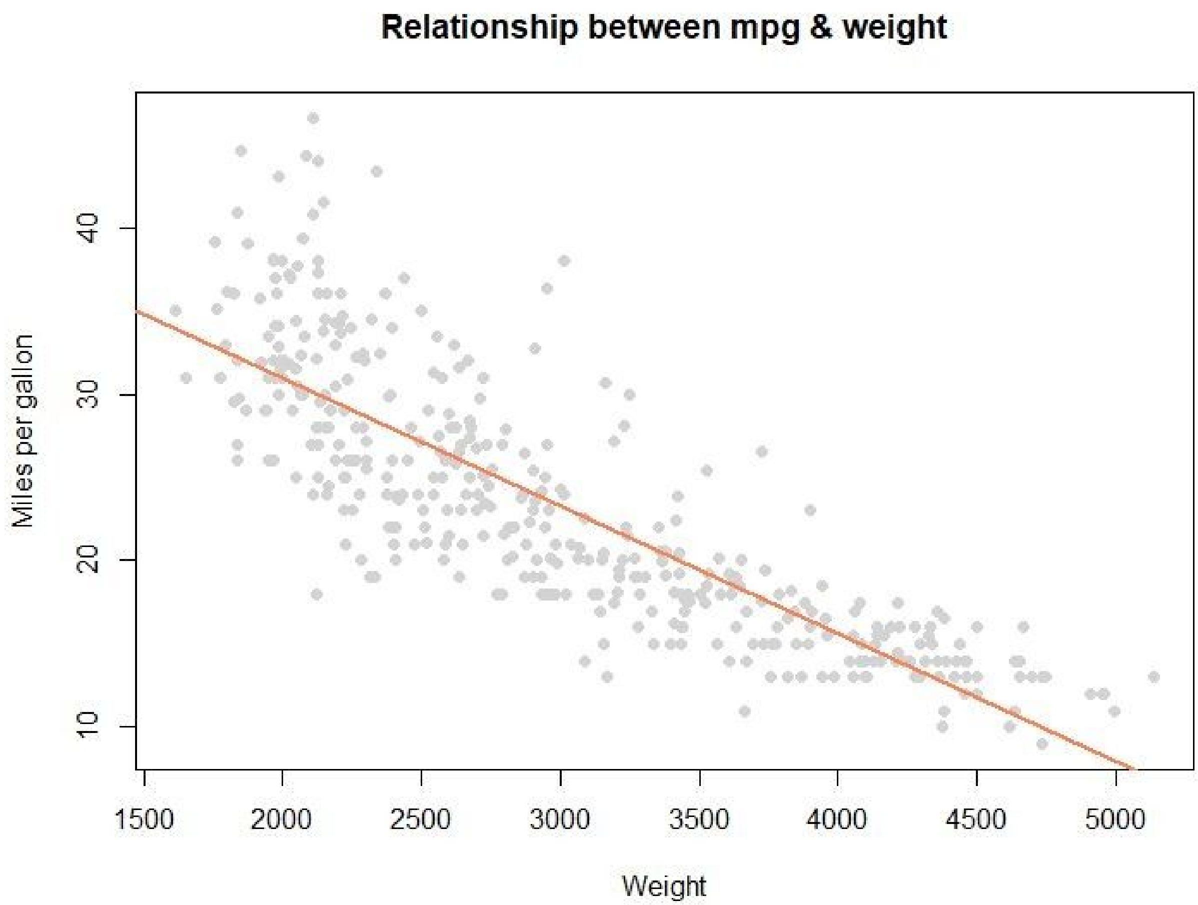
relation1 <- lm(mpg ~ horsepower , data = data)
plot(data$mpg ~ data$horsepower, col="lightgray", main="Relationship between mpg & horsepower", xlab="Horsepower", ylab="Miles per gallon", pch=16)
abline(relation1, col = "coral" , lwd = 2.5)



from the plots we can say that vehicles with more horsepower tend to have high fuel economy i.e mpg

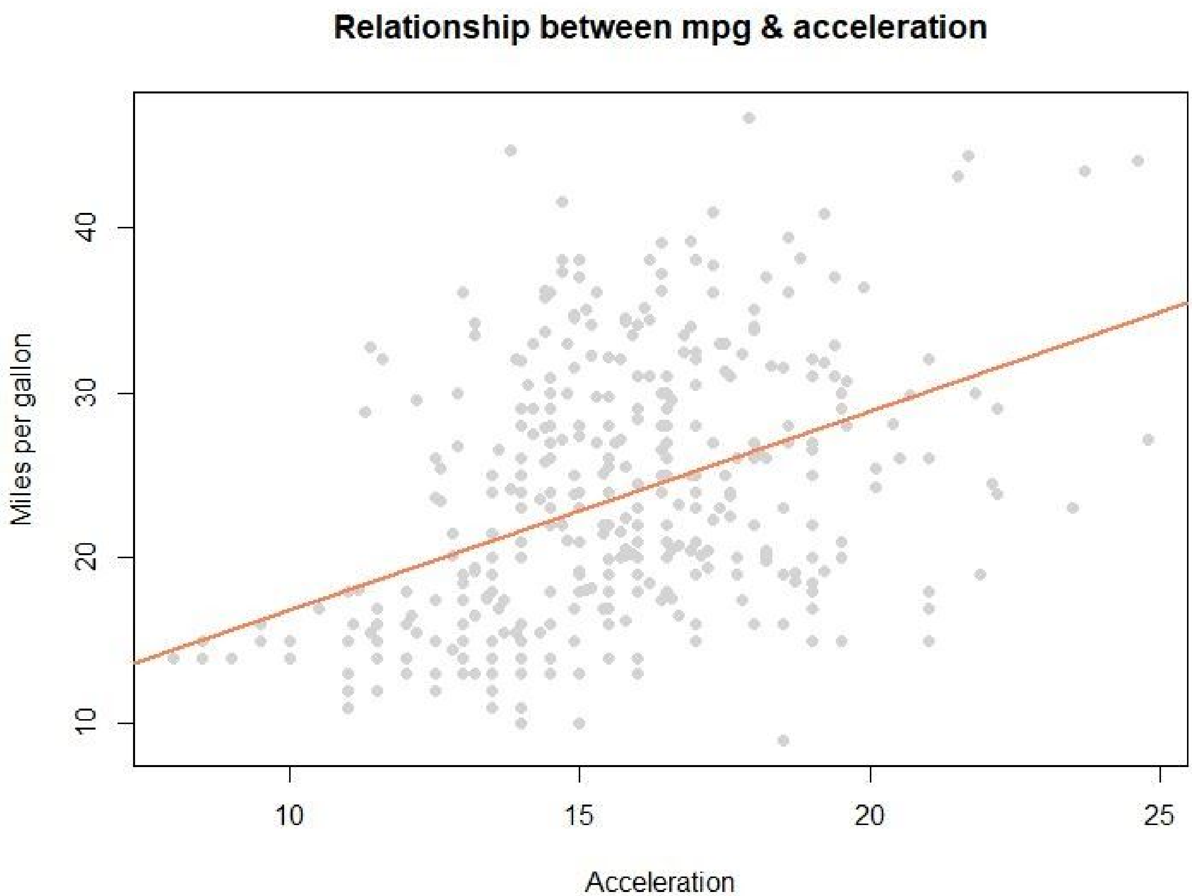
lets analyze the relation between weight and miles per gallon

```
relation2 <- lm(mpg ~ weight , data = data)
plot(data$mpg ~ data$weight, col="lightgray", main="Relationship between mpg & weight", xlab="Weight", ylab="Miles per gallon", pch=16)
abline(relation2, col = "coral" , lwd = 2.5)
```



from the plot we can say that as weight of vehicle increases mpg decreases.
this means that heavy vehicles are not fuel efficient.

```
relation3 <- lm(mpg ~ acceleration , data = data)
plot(data$mpg ~ data$acceleration, col="lightgray", main="Relationship between mpg & acceleration", xlab="Acceleration", ylab="Miles per gallon", pch=16)
abline(relation3, col = "coral" , lwd = 2.5)
```

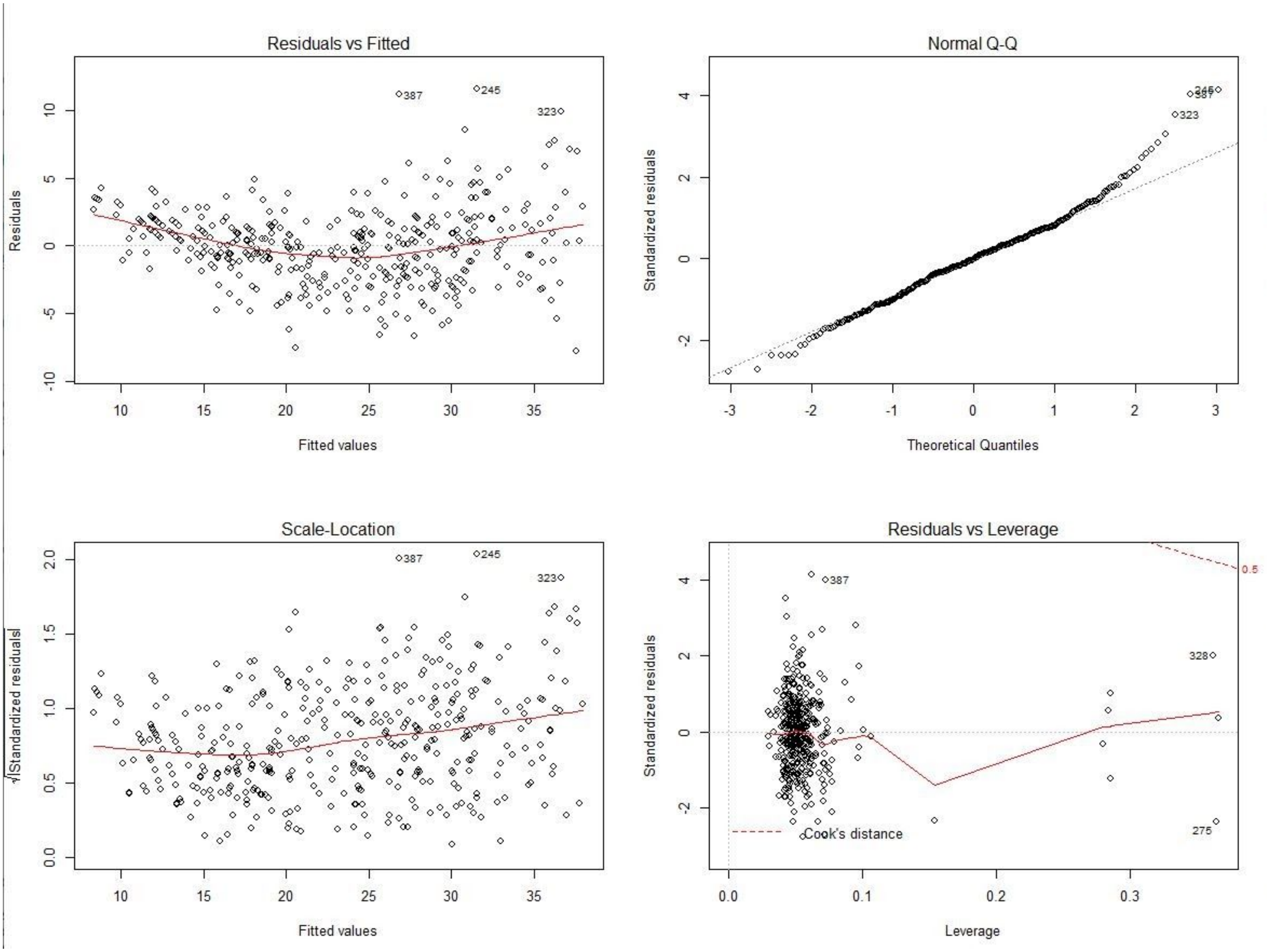


from the plots we see that miles per gallon increseases as acceleration increases.

 # g

By looking at the plots we created in part f, we can say that displacement, weight, horsepower, cylinders, year
acceleration,year can be used to predict the mpg. But not sure.
lets use regression to determine best predictors

```
model<-lm(mpg~cylinders+displacement+horsepower+weight+acceleration+year+origin,data = data)
model
summary(model)

par(mfrow=c(2,2))
plot(model)
```




```
# we do see some outliers. Lets remove them and plot a new model.


outliers <- c(387, 323, 328, 275, 245)
new <- data[-outliers,]


# now only variables have highest significance i.e '***' are selected. Since They are the strongest predictors
# of the mpg. For the other factor variables we were getting high significance for only some levels. So we
# completely removed it.

model1 <- lm(mpg ~ weight+year+origin, data= new)
summary(model1)
```

Q5 Perform analysis on gun deaths in America.

```
#importing libraries
library(corrplot)
library(dplyr)
library(gmodels)
library(gplots)
library(psych)
library(corrplot) # for correlation plot
library(Hmisc)
library(ggplot2) # for plots
library(ggthemes)
library(feather)

dataset = read.csv("gun_deaths.csv")
dataset = na.omit(dataset)
dummy = dataset
```

#A

#To determine the number of deaths by month we first need to convert the months into qualitative.

```
dataset$month <- as.factor(dataset$month)
new <- data.frame(summary(dataset$month))
```

#code output

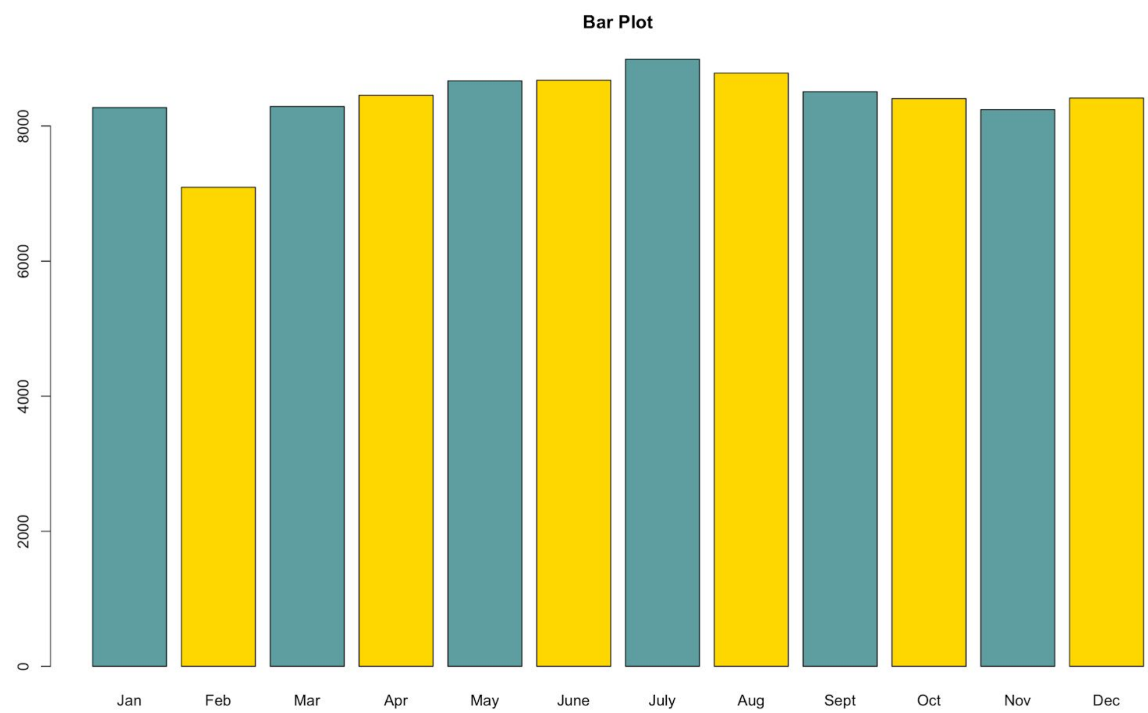
```
# 1    2    3    4    5    6    7    8    9   10   11   12
# 8273 7093 8289 8455 8669 8677 8989 8783 8508 8406 8243 8413
```

#B

#Let's relabel the month to its corresponding name as January to December. It stores the month name for each corresponding month.

```
dataset$month<- factor(dataset$month,
   levels = c(1,2,3,4,5,6,7,8,9,10,11,12),
   labels = c("Jan", "Feb", "Mar","Apr","May","June","July","Aug","Sept","Oct","Nov","Dec"))
tab <- table(dataset$month)
```

```
# Let's add margins for a better picture
ptab <-addmargins(prop.table(tab))
addmargins(prop.table(tab))
barplot(tab, main = "Bar Plot", col=c("cadetblue", "gold"))

# now let's create a bar plot
options(scipen = 99)

x <- table(as.factor(dataset$id),dataset$label)
barplot(x)
```



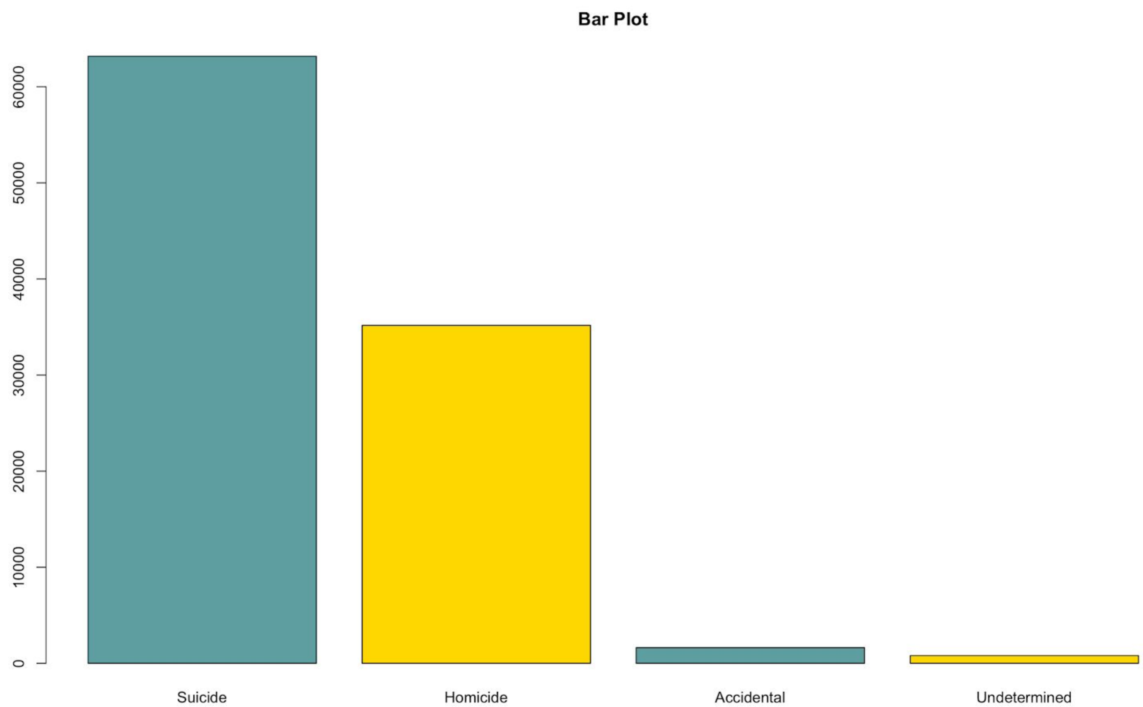

```
dev.off()
asa
```

#C

```
tabb <- table(dataset$intent)
tabb<-sort(tabb,decreasing = TRUE)
#code output
#Suicide    Homicide   Accidental Undetermined
# 62291     33329      1598       797

#Let's add margins for a better picture
ptab <-addmargins(prop.table(tabb))
```

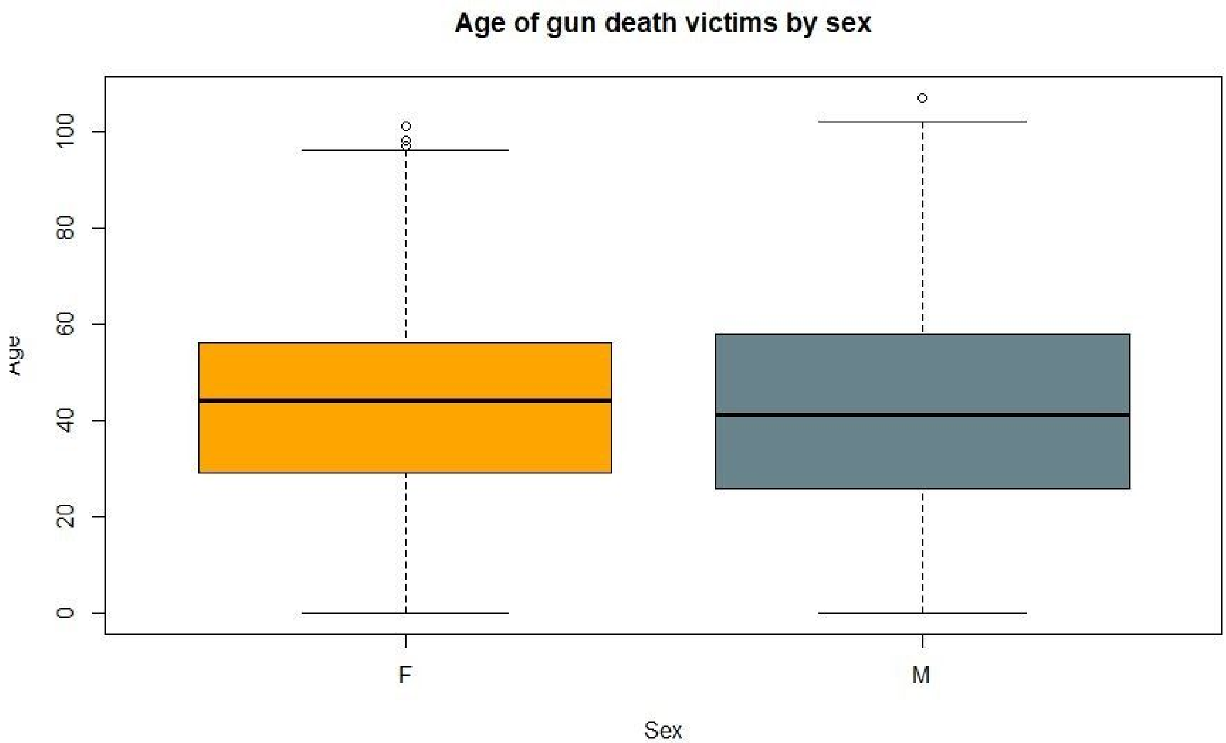
#code output

```
#Suicide      Homicide     Accidental    Undetermined     Sum
#0.635525175  0.340039790  0.016303627  0.008131408  1.000000000
addmargins(prop.table(tabb))
barplot(tabb, main = "Bar Plot", col=c("cadetblue", "gold"))
```



 # D

```
boxplot(dataset$age ~dataset$sex, data=dataset, main="Age of gun death victims by sex", xlab="Sex", ylab="Age",col=c("orange", "lightblue4"))
#The following line of code gives us only gun deaths of females.

female <- dataset[ which(dataset$sex=='F'), ]

# now lets calculate average age of female gun deaths
female <- na.omit(female)
ans  <- mean(female$age)

summary(dataset$education)
```



#E

ans <- count(dataset[which(dataset$education=='HS/GED' & dataset$race =='White' & dataset$year == 2012 & dataset$sex == "M"),])

#In 2012, 7794 there were 7994 deaths who were male, white with at least a high school education.

#F

creating a new variable named season storing the season associated with each month

dataset$season[dataset$label=="Jan" | dataset$label == "Feb" | dataset$label == "Mar"]<-"Winter"
dataset$season[dataset$label=="Apr" | dataset$label == "May" | dataset$label == "Jun"]<-"Spring"
dataset$season[dataset$label=="Jul" | dataset$label == "Aug" | dataset$label == "Sept"]<-"Summer"
dataset$season[dataset$label=="Oct" | dataset$label == "Nov" | dataset$label == "Dec"]<-"Fall"

table(dataset$season)
#code output

Fall Spring Summer Winter
24388 25045 25548 23034

from the table it is clear that Summer is the season with most gun deaths.

 # G

Are whites who are killed by guns more likely to die because of suicide or homicide? How does this compare to blacks and Hispanics?

#Case A: *Whites*
count(dataset[which(dataset$intent=='Suicide' & dataset$race =='White'),])
#56415
count(dataset[which(dataset$intent=='Homicide' & dataset$race =='White'),])
#8293

56415/98015 # i.e. 57.5%
8293/98015 # i.e 8.4 %

#We see from here that in case of white, gun violences involving *suicide* are more likely to happen.

#Case B: *Blacks*

count(dataset[which(dataset$intent=='Suicide' & dataset$race =='Black'),])
# 3285
count(dataset[which(dataset$intent=='Homicide' & dataset$race =='Black'),])
# 18956

3285/98015 # i.e. 3.35%
19510/98015 # i.e 19.91%

#Therefore, in case of blacks, gun violences involving *homicide* are more likely to happen.

#Case C: *Hispanics*

count(dataset[which(dataset$intent=='Suicide' & dataset$race =='Hispanic'),])
# 3120
count(dataset[which(dataset$intent=='Homicide' & dataset$race =='Hispanic'),])
# 5269

3120/98015 # i.e. 3.1%

5634/98015 # i.e 5.7%

#We see that for the case of Hispanics, gun violences involving *homicide* are more likely to happen.

 # H

we can see that variable police are numeric which does not make sense. It should be factor.

dataset$police <- as.factor(dataset$police)

Now in order to check whether the police involved gun deaths are significantly different from other gun
deaths, we will use the chi square test. The reason being since both intent and police are factors.

table(dataset$police)
#code output
```
#     0       1
# 97996   19
```
The police involved gun deaths are very less as compared to the non-police involvement.

Assessing relationships between police involvement and other variables.

 # with intent

table(dataset$police,dataset$intent)
#code output
```
#Accidental Homicide Suicide Undetermined
  0    1598   33310     62291        797
  1       0      19         0          0
```

from the table we see all the gundeaths in which police were involved are homicide.Same is not the case when police were not involved.

 # with year

table(dataset$year,dataset$police)
#code output
```
 #          0       1
 #2012 32608    7
```

```
 #2013 32723    7
 #2014 32665    5
```

from this we can see that from 2012 to 2014 non police involvement gun violences were more than police involved
gun violences.

 # with sex

```
table(dataset$sex,dataset$police)
```

#code output

```
#      0        1
#  F 14180     0
#  M 83816     19
```

From this we get the information that all police invloved gun deaths involved males.
Whereas in non police involved gun deaths involving males was almost six times more as compared to females

```
table(dataset$race,dataset$police)
```

#code output

```
#
#                                       0        1
# Asian/Pacific Islander              1261      0
#  Black                              22672    3
#  Hispanic                           8601      2
#  Native American/Native Alaskan     878        0
#  White                              64584    14
```

In case of police involvement of non-police involvement, majority of victims belonged to White race.
with education

```
table(dataset$education,dataset$police)
```

#code output

```
#
                    0         1
#  BA+             12879     0
#  HS/GED          42247    11
#  Less than HS    21444     4
#  Some college    21426     4
```

In both the cases, majority of the victims were at least High School graduates.

#with age

aggregate(dataset$age, by = list(dataset$police) ,FUN = mean)

In police involved gun deaths the average age of victims is ~ 36 and for non police invloved is ~ 44.

 # with place

table(dataset$place,dataset$police)

#code output

```
#
#                              0       1
#  Farm                        465     0
#  Home                        59619   3
#  Industrial/construction     239     0
#  Other specified             13543   2
#  Other unspecified           8746    8
#  Residential institution     201     0
#  School/instiution           662     0
#  Sports                      128     0
#  Street                      11003   5
#  Trade/service area          3390    1
```

police-involved gun violence were mostlty at Other unspecified place.
in non police involved gun violences were mostly at Home.

Since all the police involved gun violences were Homicide, so keeping homicide category aside they are different
from other intent types.

Q6 Perform analysis on salary class dataset

```
install.packages("arules")
install.packages("arulesViz")

library(rpart)
library(party)
library(readxl)
library(rattle)
library(rpart.plot)
library(RColorBrewer)
library(arules)
library(arulesViz)
```

 # a

```
# Importing the dataset

dataset <- read_excel("salary-class.xlsx")

# substituting all "?" with NA and then removing it.

dataset$AGE[dataset$AGE == "?"] <- NA
dataset$EMPLOYER[dataset$EMPLOYER == "?"] <- NA
dataset$DEGREE[dataset$DEGREE == "?"] <- NA
dataset$MSTATUS[dataset$MSTATUS == "?"] <- NA
dataset$JOBTYPE[dataset$JOBTYPE == "?"] <- NA
dataset$SEX[dataset$SEX == "?"] <- NA
dataset$`C-GAIN`[dataset$`C-GAIN` == "?"] <- NA
dataset$`C-LOSS`[dataset$`C-LOSS` == "?"] <- NA
dataset$HOURS[dataset$HOURS == "?"] <- NA
dataset$COUNTRY[dataset$COUNTRY == "?"] <- NA
dataset$INCOME[dataset$INCOME == "?"] <- NA
dataset$EMPLOYER[dataset$EMPLOYER == "?"] <- NA
dataset <- na.omit(dataset)

dataset$MSTATUS <- as.factor(dataset$MSTATUS)
dataset$EMPLOYER <- as.factor(dataset$EMPLOYER)
dataset$DEGREE <- as.factor(dataset$DEGREE)
dataset$JOBTYPE <- as.factor(dataset$JOBTYPE)
dataset$SEX <- as.factor(dataset$SEX)
dataset$COUNTRY <- as.factor(dataset$COUNTRY)
```

dataset$INCOME <- as.factor(dataset$INCOME)

Spliting the dataset 60%-40%.

set.seed(1234)
ind <- sample(2, nrow(dataset), replace=TRUE, prob=c(0.6, 0.4))
trainData <- dataset[ind==1,]
testData <- dataset[ind==2,]

 # b

Method = class for classification tree and anova for regression tree.

rfit = rpart(INCOME ~ AGE + EMPLOYER + DEGREE + MSTATUS + JOBTYPE + SEX
    + `C-GAIN` + `C-LOSS` + HOURS + COUNTRY, data = trainData,
    method = "class",parms = list(split = "gini"),control = rpart.control(cp = 0.01))
rpart.plot(rfit, type = 3, cex = 0.4)

printcp(rfit)
#code output
#Classification tree:

```
#rpart(formula = INCOME ~ AGE + EMPLOYER + DEGREE + MSTATUS +
#       JOBTYPE + SEX + `C-GAIN` + `C-LOSS` + HOURS + COUNTRY, data = trainData,
#      method = "class", parms = list(split = "gini"), control = rpart.control(cp = 0.01))

#Variables actually used in tree construction:
#  [1] C-GAIN  C-LOSS  DEGREE  JOBTYPE MSTATUS

#Root node error: 4536/17985 = 0.25221

#n= 17985

#CP nsplit rel error  xerror     xstd
#1 0.131944      0   1.00000 1.00000 0.012840
#2 0.039683      2   0.73611 0.75728 0.011622
#3 0.034612      3   0.69643 0.71098 0.011342
#4 0.014440      4   0.66182 0.65564 0.010984
#5 0.010000      7   0.61640 0.63757 0.010861

pred_Test_class <- predict(rfit, newdata = testData, type = "class")
mean(pred_Test_class != testData$INCOME)
#0,152
summary(rfit)
# the leaf nodes can be determined by viewing the output with *.
# In all there eight leaves in the tree.
```

 # c

Looking at the Variable importance we can say that MSTATUS, JOBTYPE and C-GAIN are the major predictors of the variable
We get this information from the Variable importance field in the summary(rfit)

 # d

```
install.packages("tidyrules")
library("tidyrules")
rules <- tidyRules(rfit)

rules

# For > 50K
#  MSTATUS %in% c('Divorced', 'Married-spouse-absent', 'Never-married', 'Separa~ >50K
161     0.982 3.89
```

```
# MSTATUS %in% c('Married-AF-spouse', 'Married-civ-spouse') & JOBTYPE %in% c('~
>50K     188     0.974 3.86
# MSTATUS %in% c('Married-AF-spouse', 'Married-civ-spouse') & JOBTYPE %in% c('~
>50K     146     0.973 3.86

# For <= 50k
#    MSTATUS %in% c('Divorced', 'Married-spouse-absent', 'Never-married', 'Separa~
<=50K   9390     0.951 1.27
# the other two rules don't meet the criteria.
# so the best two rules are as follows
# MSTATUS %in% c('Married-AF-spouse', 'Married-civ-spouse') & JOBTYPE %in% c('~
<=50K   4125     0.740 0.989
# MSTATUS %in% c('Married-AF-spouse', 'Married-civ-spouse') & JOBTYPE %in% c('~
<=50K   1722     0.560 0.749
```

 # e

```
# Second decision tree
# We are not pruning this tree, allowing it to grow

rfit1 = rpart(INCOME ~ AGE + EMPLOYER + DEGREE + MSTATUS + JOBTYPE + SEX
          + `C-GAIN` + `C-LOSS` + HOURS + COUNTRY, data = trainData,
            method = "class",parms = list(split = "gini"),control = rpart.control(minsplit = 0,
minbucket = 0 , cp = 0))
rpart.plot(rfit)
printcp(rfit1)
pred_Test_class <- predict(rfit1, newdata = testData, type = "class")


# Third decision tree
# We are asigning 500 records to the parent branch and 100 records to the child branch.
rfit2 = rpart(INCOME ~ AGE + EMPLOYER + DEGREE + MSTATUS + JOBTYPE + SEX
          + `C-GAIN` + `C-LOSS` + HOURS + COUNTRY, data = trainData,
          method = "class",parms = list(split = "gini"),control = rpart.control(minsplit = 500,
minbucket = 100 , cp = 0.01))

rpart.plot(rfit2)
printcp(rfit2)
pred_Test_class <- predict(rfit2, newdata = testData, type = "class")
```

```
mean(pred_Test_class != testData$INCOME)
rfit2$cptable

# Calculating accuracies for the trees

pred_Test_class <- predict(rfit, newdata = testData, type = "class")
pred_Train_class <- predict(rfit, newdata = trainData, type = "class")

table(pred_Test_class,testData$INCOME)
```

#code output

```
#pred_Test_class <=50K >50K
#             <=50K   8837 1483
#              >50K     368  1489

# testing set accuracy is 84.79%
table(pred_Train_class,trainData$INCOME)
```

#code output

```
#pred_Train_class <=50K  >50K
#                <=50K 12945  2292
#                 >50K   504  2244

# training set accuracy is 84.45%

pred_Test_class_1 <- predict(rfit1, newdata = testData, type = "class")
pred_Train_class_1 <- predict(rfit1, newdata = trainData, type = "class")

table(pred_Test_class_1,testData$INCOME)
```

#code output

```
#pred_Test_class_1 <=50K >50K
#                 <=50K  8004 1104
#                  >50K   1201 1868

# testing set accuracy is 81%
table(pred_Train_class_1,trainData$INCOME )
```

#code output

```
#pred_Train_class_1 <=50K  >50K
#                 <=50K 13294   292
#                  >50K    155  4244

# training set accuracy is 97.5%

pred_Test_class_2 <- predict(rfit2, newdata = testData, type = "class")
pred_Train_class_2 <- predict(rfit2, newdata = trainData, type = "class")

table(pred_Test_class_2,testData$INCOME)
```

#code output

```
#pred_Test_class_2 <=50K >50K
#               <=50K  8837 1483
#                >50K   368 1489

# testing set accuracy is 84.79%
table(pred_Train_class_2,trainData$INCOME )
```

#code output

```
#pred_Train_class_2 <=50K  >50K
#                 <=50K 12945  2292
#                  >50K    504  224
# training set accuracy is 84.45%

# for the third decision tree lets take cp = 0.131 and again plot the tree.

rfit2_new = rpart(INCOME ~ AGE + EMPLOYER + DEGREE + MSTATUS + JOBTYPE + SEX
        + `C-GAIN` + `C-LOSS` + HOURS + COUNTRY, data = trainData,
           method = "class",parms = list(split = "gini"),control = rpart.control(minsplit =
500, minbucket = 100 , cp = 0.131))
pred_Test_class_2_new <- predict(rfit2_new, newdata = testData, type = "class")
pred_Train_class_2_new <- predict(rfit2_new, newdata = trainData, type = "class")
table(pred_Test_class_2_new,testData$INCOME)
```

#code output

```
#pred_Test_class_2_new <=50K >50K
```

```
#                              <=50K  8143 1275
#                                >50K   1062 1697

# testing set accuracy is 80.8%
table(pred_Train_class_2_new,trainData$INCOME )
```

#code output

```
#pred_Train_class_2_new <=50K  >50K
#                                <=50K  11987  1877
#                                  >50K   1462  2659

# training set accuracy is 81.43%

# for the third decision tree lets take cp = 0.039
rfit2_new_1 = rpart(INCOME ~ AGE + EMPLOYER + DEGREE + MSTATUS + JOBTYPE + SEX
                + `C-GAIN` + `C-LOSS` + HOURS + COUNTRY, data = trainData,
                    method = "class",parms = list(split = "gini"),control = rpart.control(minsplit = 500, minbucket = 100 , cp = 0.039))
pred_Test_class_2_new_1 <- predict(rfit2_new_1, newdata = testData, type = "class")
pred_Train_class_2_new_1 <- predict(rfit2_new_1, newdata = trainData, type = "class")

table(pred_Test_class_2_new_1,testData$INCOME)

#code output

#pred_Test_class_2_new_1 <=50K >50K
#                                    <=50K  8139 1159
#                                      >50K   1066 1813

# testing set accuracy is 81.72%
table(pred_Train_class_2_new_1,trainData$INCOME)

#code output

#pred_Train_class_2_new_1 <=50K  >50K
#                           <=50K          11983  1693
#                             >50K           1466  2843

# training set accuracy is 82.43%
```

What do we infer?
The three trees differ briefly when it comes to overfitting. Since for the second decision tree i.e one with no pruning
has accuracy of about ~97% when it comes to training set. But falls drastically when we check it for testing set.
So if we do not perform pruning at all then overfitting occurs.

Decision tree with no pruning is the most accurate on the training data.
Decision trees i.e default and the third one when we keep cp = 0.01 is the most accurate on the testing data.