1 (a) Considering "buy-computer" as the target variable, which of the attributes would you select as the root in a decision tree that is constructed using the gain ratio impurity measure?

We need to use the following parameters to determine the Gain Ratio:
Information gain, split info and gain ratio
The formulae is given by:

Information Gain = entropy(parent) –[average entropy(children)]
GainT,A=infoT– Σ v∈values(A)TvTinfo(Tv)

SplitInfo(T) = −Σ(|Ti|/|T|) log2(|Ti|/|T|)
GainRatio(T) = Gain (T)/SplitInfo(T)
info(T)=−p+log2p+ −p−log2p−

<u>Income:</u>

Info(T)	-7/14 log $_2$ (7/14) - 7/14 log $_2$ (7/14)	1
Info(high)	-0/3 log $_2$ (0/3) - 3/3 log $_2$ (3/3)	0
Info(medium)	-4/6 log $_2$ (4/6) - 4/6 log $_2$ (4/6)	0.9146
Info(low)	-3/5 log $_2$ (3/5) - 3/5 log $_2$ (3/5)	0.9708
Info(income)	0+ (6/14) * 0.9146 + (5/14) * 0.9708	0.73868
Gain(income)	1 – 0.7403	0.26132
Splitinfo (income)	-(3/14) log$_2$(3/14) - (5/14) log$_2$(5/14) - (6/14) log$_2$(6/14)	1.5305
Gain Ratio (Income)	0.26132/1.5305	0.1707

<u>Student</u>

Info(T)	-7/14 log $_2$ (7/14) - 7/14 log $_2$ (7/14)	1
Info(yes)	-5/8 log $_2$ (5/8) - 3/8 log $_2$ (3/5)	0.9543
Info(no)	-4/6 log $_2$ (4/6) - 4/6 log $_2$ (4/6)	0.9182
Info(student)	-3/5 log $_2$ (3/5) - 3/5 log $_2$ (3/5)	0.9388
Gain(student)	0+ (6/14) * 0.9146 + (5/14) * 0.9708	0.0611
Splitinfo (student)	1 – 0.7403	0.9851
Gain Ratio (student)	-(3/14) log$_2$(3/14) - (5/14) log$_2$(5/14) - (6/14) log$_2$(6/14)	0.0620

<u>Credit rating</u>

Info(T)	-7/14 log $_2$ (7/14) - 7/14 log $_2$ (7/14)	1
Info(fair)	-2/7 log $_2$ (2/7) - 5/7 log $_2$ (5/7)	0.8631
Info(excellent)	-5/7 log $_2$ (5/7) - 4/6 log $_2$ (4/6)	0.8631
Info(Credit Rating)	(7/14) * 0.8631 + (7/14) * 0.8631	0.863
Gain(Credit Rating)	1 – 0.8631	0.1369
Splitinfo (Credit Rating)	-(7/14)log2(7/14)-(7/14)log2(7/14)	1
Gain Ratio (Credit Rating)	0.1368/1	0.1369

Thus for the initial Branch we have:

Income		Student		Credit Rating	
Info	0.73868	Info	0.9388	Info	0.863
Gain	0.26132	Gain	0.0611	Gain	0.1369
Split Info	1.5305	Split Info	0.9851	Split Info	1
Gain Ratio	0.1707	Gain Ratio	0.0620	Gain Ratio	0.1369

From the Table above it is evident that Income attribute has the highest Gain Ratio. So we select Income as the Root Node.

b) For the same data set, suppose we decide to construct a decision tree using the Gini index impurity measure. Which attribute would be the best option to use as the root node?

Gini: $1-(P_+)^2-(P_-)^2$

Gini Index of Income:

Gini(high)	$1-((0/3)^2+(3/3)^2)$	0
Gini(medium)	$1-((4/6)^2+(2/6)^2)$	0.4444
Gini(low)	$1-((3/5)^2+(2/5)^2)$	0.48
Gini Index(Income)	(3/14)(0)+ 6/14)(0.44) +(5/14)(0.48)	0.3618

Gini Index of Student:

Gini(Yes)	$1-((5/8)^2+(3/8)^2)$	0.46875
Gini(No)	$1-((2/6)^2+(4/6)^2)$	0.4444
Gini Index(Student)	(8/14)(0.46875)+(6/14)(0.444)	0.4581

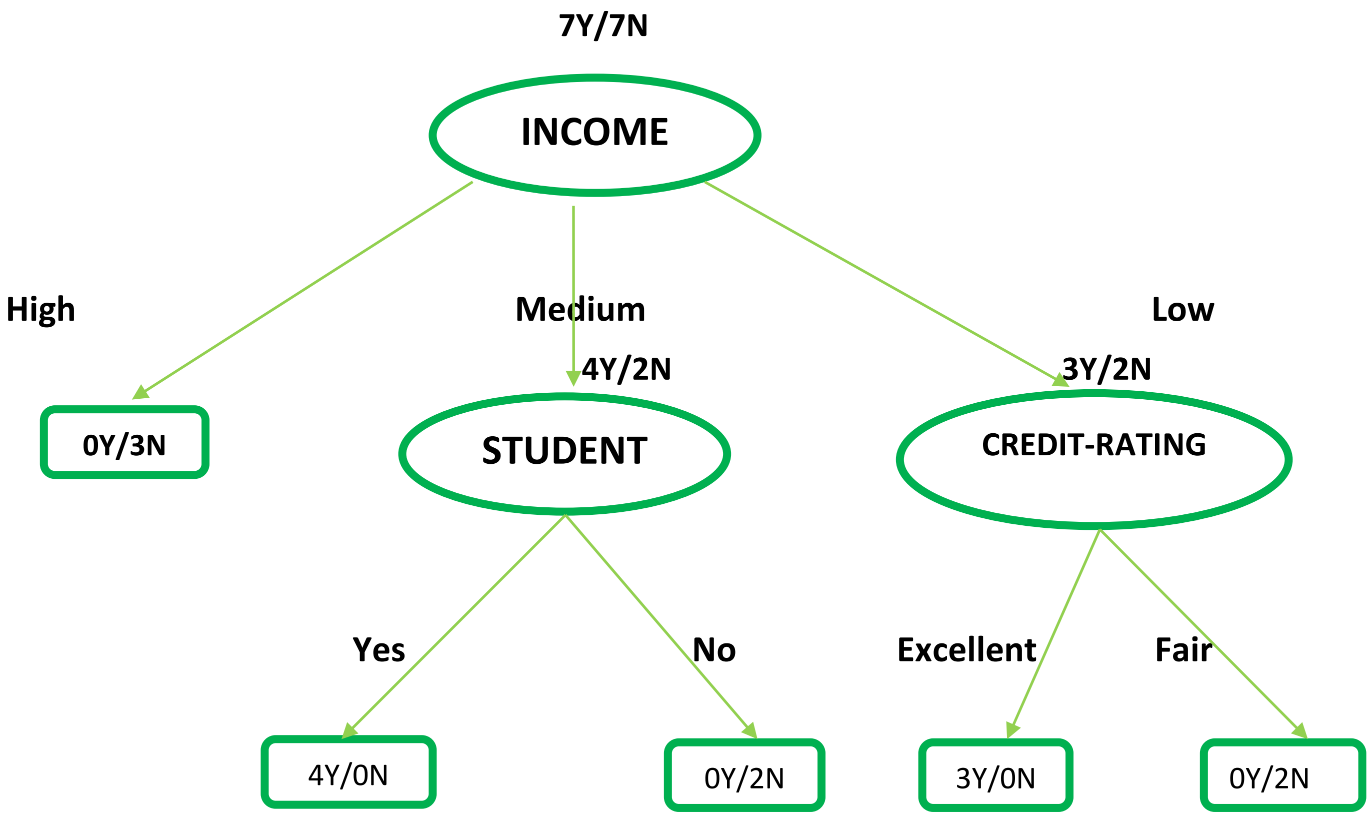
Gini Index of Credit Rating:

Gini(Fair)	$1-((2/7)^2+(5/7)^2)$	0.4081
Gini(Excellent)	$1-((5/7)^2+(2/7)^2)$	0.4081
Gini Index(Credit)	(7/14)(0.408)+(7/14)(0.408)	0.4081

	Income	Student	Credit Rating
Gini Index	0.3618	0.4581	0.4081

Out of three attributes, Income has the lowest Gini Index and hence Income is selected as the Root Node.

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



1 d) Using your decision tree, provide two decision rules that we can use to predict whether a student is going to buy computer or not. Justify your choice.

If Income is High, then buys-computer is No
Support = 3/14 confidence = 100 %
If Income is Medium, student is Yes then buys-computer is Yes
Support = 4/14 confidence = 100 %
If Income is Medium, Student is No then buys-computer is No
Support = 2/14 confidence = 100 %
If Income is Low, Credit-Rating is Fair then buys-computer is No
Support = 2/14 confidence = 100 %
If Income is Low, Credit-Rating is Excellent then buys-computer is Yes
Support = 3/14 confidence = 100 %

Problem 2. Given the dataset in the following table, use the Naive Bayes classifier to classify the new

ID	a_1	a_2	a_3	True Class
1	T	T	5.0	Y
2	T	T	7.0	Y
3	T	F	8.0	N
4	F	F	3.0	Y
5	F	T	7.0	N
6	F	T	4.0	N
7	F	F	5.0	N
8	T	F	6.0	Y
9	F	T	1.0	N

X = (a_1 = T, a_2 = F, a_3 = 1.0)

P(a_1 = T | TC = Yes) = 3/4

P(a_2 = F | TC = Yes) = 2/4

P(a_3 = 1.0 | TC = Yes) = $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}}$ mean = 5.25, standard deviation = 1.71

$= \frac{1}{\sqrt{2\pi 1.71^2}} e^{-\frac{(1.0-5.25)^2}{2*1.71*1.71}}$ = 0.0107

P(Yes) x P(X | TC = Yes) = P(a_1 = T | TC = Yes) x P(a_2 = F | TC = Yes) x P(a_3 = 1.0 | TC = Yes) x P(Yes)

= 3/4 x 2/4 x 0.0107 x 4/9 = **0.0018**

P(a_1 = T | TC = No) = 1/5

P(a_1 = F | TC = No) = 2/5

P(a_3 = 1.0 | TC = No) = $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-u)^2}{2\sigma^2}}$ mean = 5, standard deviation = 2.74

$= \frac{1}{\sqrt{2\pi 2.74^2}} e^{-\frac{(1.0-5)^2}{2*2.74*2.74}}$ = 0.0502

P(No) x P(X | TC = No) = P(a_1 = T | TC = No) x P(a_2 = F | TC = No) x P(a_3 = 1.0 | TC = No) x P(No)

= 1/5 x 2/5 x 0.0502 x 5/9 = **0.002**

Comparing the resulting probabilities for True Class Yes and No, we classify the new point as **"No"**, since it's probability is higher

Problem 3.

libraries required for this problem:

library(psych)

library(corrplot)
library(dplyr)
library(car)
library(gplots)

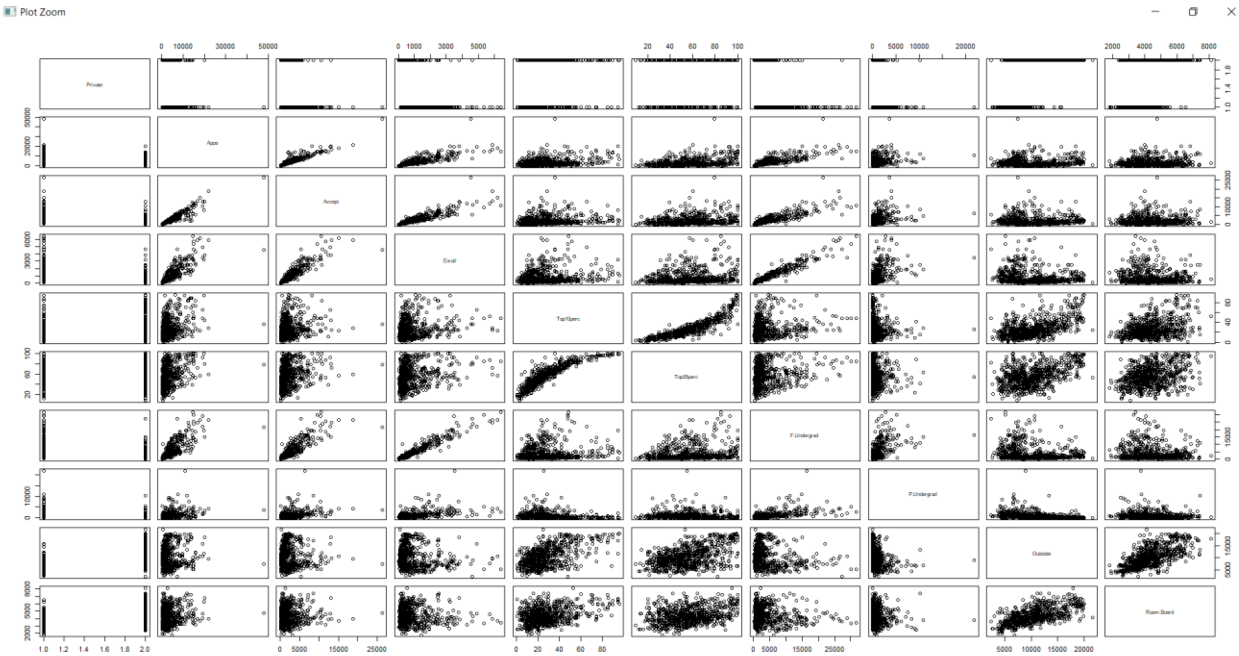
3.
(a) Read the data into R. Call the loaded data "college".
college <- read.csv(file.choose(), header = T)
View(college)
rownames (college) <- college[,1]
View(college)

3.(b)Write a code to eliminate the 1st column
college[,1]<-NULL
View(college)
str(college)

3.c) Provide a summary statistics for numerical variables in the data set.
describe(college[,2:18])
Selecting all numeric variables except Private which is a factor

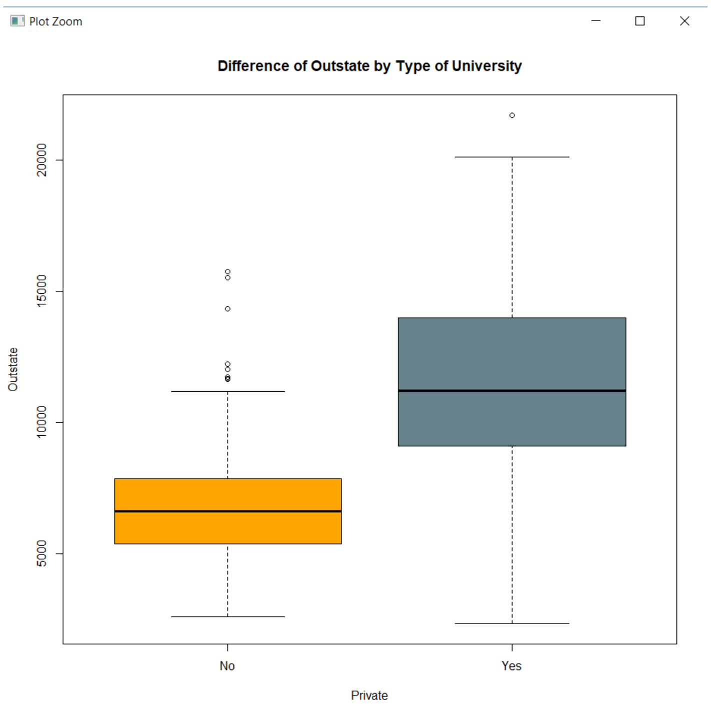
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
Apps	1	777	3001.64	3870.20	1558.0	2193.01	1463.33	81.0	48094.0	48013.0	3.71	26.52	138.84
Accept	2	777	2018.80	2451.11	1110.0	1510.29	1008.17	72.0	26330.0	26258.0	3.40	18.75	87.93
Enroll	3	777	779.97	929.18	434.0	575.95	354.34	35.0	6392.0	6357.0	2.68	8.74	33.33
Top10perc	4	777	27.56	17.64	23.0	25.13	13.34	1.0	96.0	95.0	1.41	2.17	0.63
Top25perc	5	777	55.80	19.80	54.0	55.12	20.76	9.0	100.0	91.0	0.26	-0.57	0.71
F.Undergrad	6	777	3699.91	4850.42	1707.0	2574.88	1441.09	139.0	31643.0	31504.0	2.60	7.61	174.01
P.Undergrad	7	777	855.30	1522.43	353.0	536.36	449.23	1.0	21836.0	21835.0	5.67	54.52	54.62
Outstate	8	777	10440.67	4023.02	9990.0	10181.66	4121.63	2340.0	21700.0	19360.0	0.51	-0.43	144.32
Room.Board	9	777	4357.53	1096.70	4200.0	4301.70	1005.20	1780.0	8124.0	6344.0	0.48	-0.20	39.34
Books	10	777	549.38	165.11	500.0	535.22	148.26	96.0	2340.0	2244.0	3.47	28.06	5.92
Personal	11	777	1340.64	677.07	1200.0	1268.35	593.04	250.0	6800.0	6550.0	1.74	7.04	24.29
PhD	12	777	72.66	16.33	75.0	73.92	17.79	8.0	103.0	95.0	-0.77	0.54	0.59
Terminal	13	777	79.70	14.72	82.0	81.10	14.83	24.0	100.0	76.0	-0.81	0.22	0.53
S.F.Ratio	14	777	14.09	3.96	13.6	13.94	3.41	2.5	39.8	37.3	0.66	2.52	0.14
perc.alumni	15	777	22.74	12.39	21.0	21.86	13.34	0.0	64.0	64.0	0.60	-0.11	0.44
Expend	16	777	9660.17	5221.77	8377.0	8823.70	2730.95	3186.0	56233.0	53047.0	3.45	18.59	187.33
Grad.Rate	17	777	65.46	17.18	65.0	65.60	17.79	10.0	118.0	108.0	-0.11	-0.22	0.62

3.(d) Use the pairs() function to produce a scatterplot matrix of the 1st ten columns or variables of the data. Recall that you can reference the 1st ten columns of a matrix A using A[,1:10].
pairs(college[,1:10])



3.(e) Use the plot() function to produce side-by-side boxplots of Outstate versus Private.

```
dev.off()
boxplot(college$Outstate ~ college$Private, data=college, main="Difference of Outstate by Type of University",
    xlab="Private", ylab="Outstate",
    col=c("orange", "lightblue4"))
```



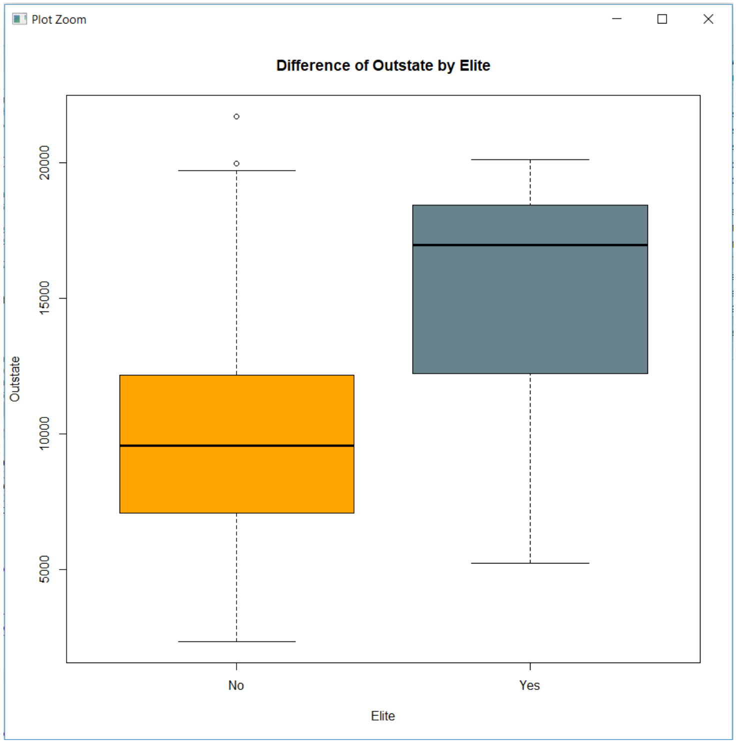
3.(f) Create a new qualitative variable, called Elite, by binning the Top10perc variable. We are going to divide universities into two groups based on whether or not the proportion of students comingfrom the top 10% of their high school classes exceeds 50%. Follow the code below.
#i. Explain each line of the above code.

Elite <- rep ("No",nrow(college))
Creating a new Qualitative variable Elite and assigning nrow(777) "No" values to it.
Elite[college$Top10perc > 50] <- "Yes" #Assigning "Yes" to all values if Elite when top10percentage column of college data exceeds 50
Elite <- as.factor(Elite)
Converting this Elite variable into a Factor
college <- data.frame(college,Elite)
Adding this Variable to the college dataframeView(college). Now Elite column has been added to college dataframe.

ii. Use the summary() function to see how many elite universities there are. Now use the #plot() function to produce side-by-side boxplots of Outstate versus Elite.

summary(college$Elite)
```
No Yes
699   78
```
There are 78 Elite universities out of 777

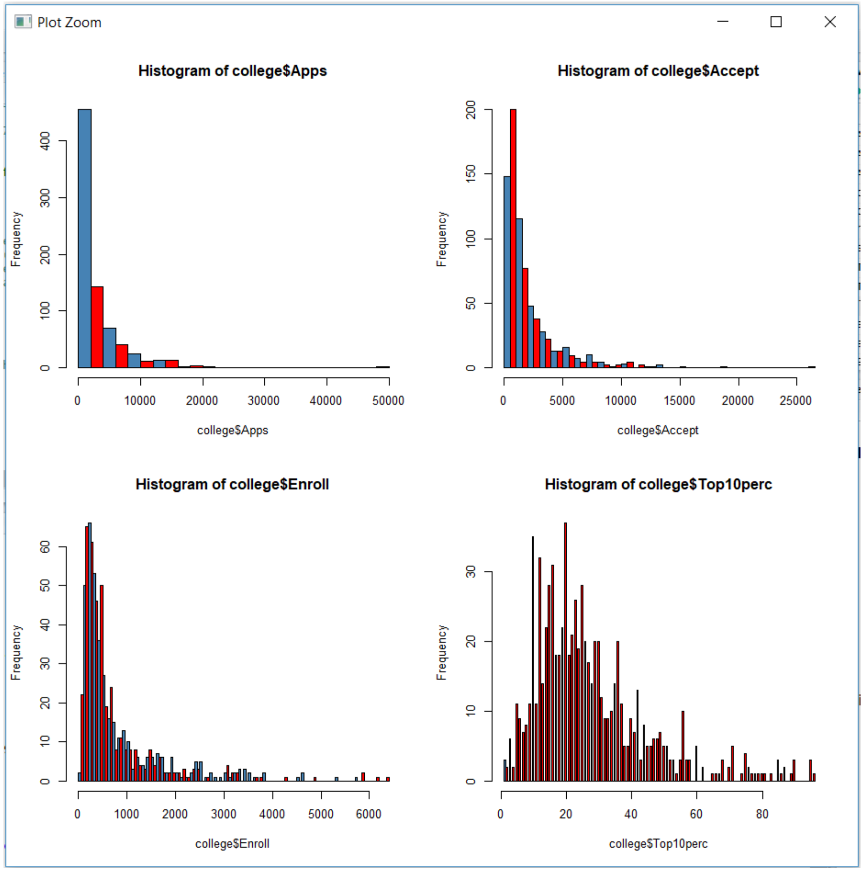
dev.off()
boxplot(college$Outstate ~ college$Elite, data=college, main="Difference of Outstate by Elite",
xlab="Elite", ylab="Outstate",
col=c("orange", "lightblue4"))



3.(g) Use the hist() function to produce some histograms with differing numbers of bins for a few of the quantitative variables. You may find the command par(mfrow=c(2,2)) useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways.

```
str(college)
dev.off()
par(mfrow=c(2,2))
hist(college$Apps, breaks=30, col=c("steelblue", "red"))
hist(college$Accept,breaks=70, col=c("steelblue", "red"))
hist(college$Enroll,breaks=100, col=c("steelblue", "red"))
hist(college$Top10perc,breaks=200, col=c("steelblue", "red"))
```

breaks has been used to create differing number of bins



Problem:4

4.(a) Remove the missing values from this data set.

Auto <- read.csv(file.choose(), header = T)

View(Auto)

str(Auto)

```
'data.frame':   397 obs. of  9 variables:
 $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
 $ cylinders   : int  8 8 8 8 8 8 8 8 8 8 ...
 $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
 $ horsepower  : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...
 $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 ...
 $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
 $ year        : int  70 70 70 70 70 70 70 70 70 70 ...
 $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161 141 54 223 241 2 ...
```

The numeric datatypes are characters so need to convert to numeric after removing "?"

Auto$horsepower <-as.numeric(sub("?", "", Auto$horsepower))

NAs are produced after removing "?"

is.na(Auto)

Auto<-na.omit(Auto)

6 NAs are removed

4.(b) Which of the predictors are quantitative, and which are qualitative? How do you check this information?

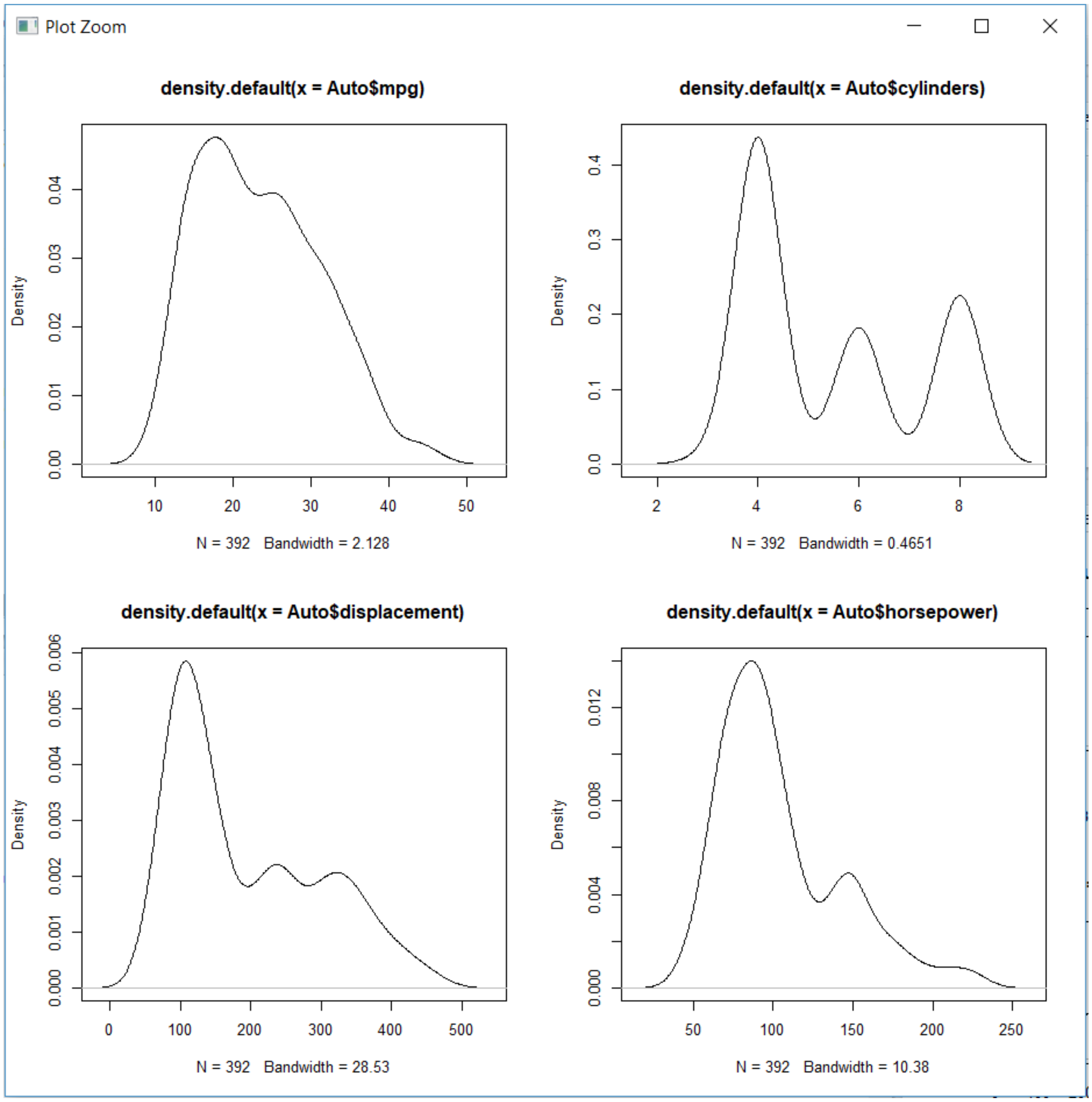
str(Auto)

plot(density(Auto$mpg)) #Quantitative - Numeric

plot(density(Auto$cylinders)) #Qualitative - Factor

plot(density(Auto$displacement)) #Quantitative - Numeric

plot(density(Auto$horsepower)) #Quantitative – Numeric



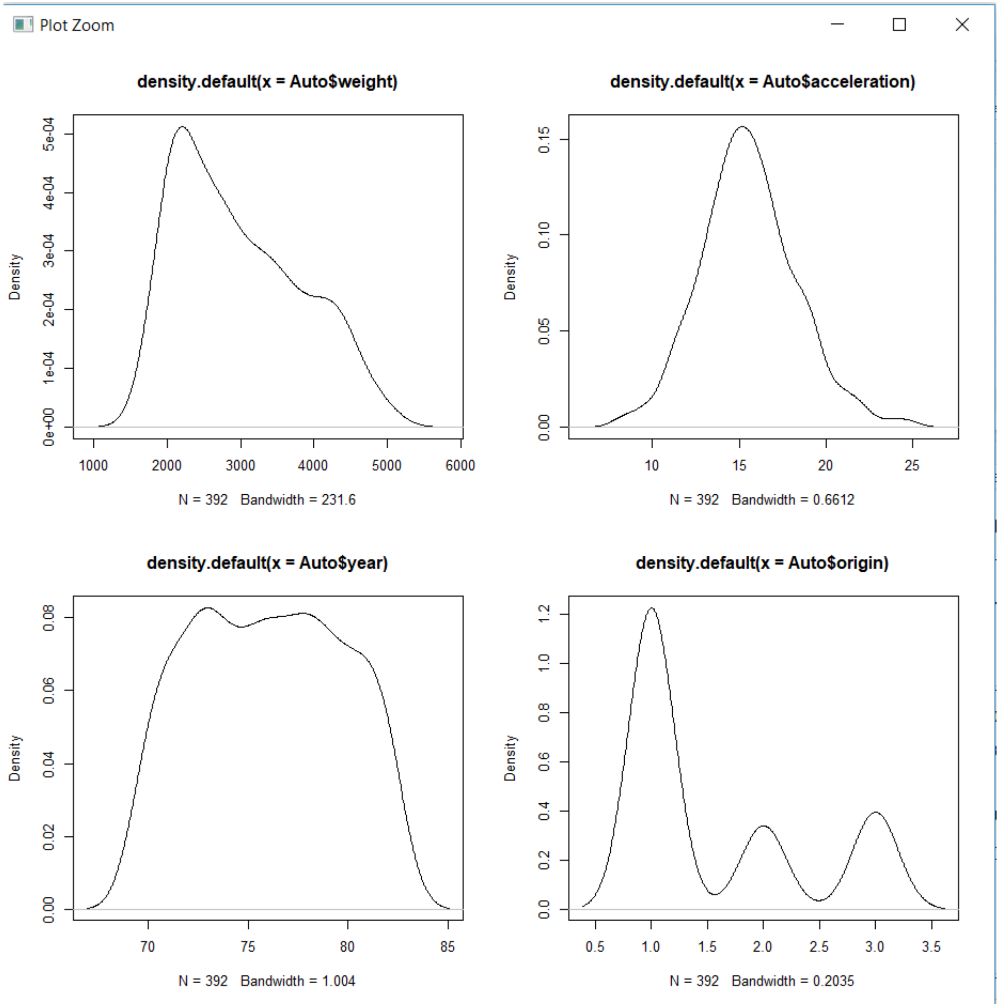
plot(density(Auto$weight)) #Quantitative - Numeric

plot(density(Auto$acceleration)) #Quantitative - Numeric

plot(density(Auto$year)) #Qualitative - Factor

plot(density(Auto$origin)) #Qualitative – Factor

plot(density(Auto$name)) #Qualitative - Factor



Auto$cylinders <- as.factor(Auto$cylinders)
Auto$year <- as.factor(Auto$year)
Auto$origin <- as.factor(Auto$origin)

We use density plot to determine whether the variable is quantitative - Numeric or Qualitative - Factors. If the density plot has multiple peaks, then it denotes there are multiple levels - thus it proves that it is a Factor. If the density plot has a single peak which resembles like a normal distribution it is a quantitative varible - Numeric

4.(c) What is the range of each quantitative predictor?

range(Auto$mpg)

range(Auto$displacement)

range(Auto$horsepower)

range(Auto$weight)

range(Auto$acceleration)

```
> range(Auto$mpg)
[1]  9.0 46.6
> range(Auto$displacement)
[1]  68 455
> range(Auto$horsepower)
[1]  46 230
> range(Auto$weight)
[1] 1613 5140
> range(Auto$acceleration)
[1]  8.0 24.8
```

#4.d) What is the mean and standard deviation of each quantitative predictor?

describe(Auto[,"mpg"])

```
describe(Auto[,"mpg"])
   vars   n  mean   sd median trimmed mad min  max range skew kurtosis   se
X1    1 392 23.45 7.81  22.75   22.99 8.6   9 46.6  37.6 0.45    -0.54 0.39
```

describe(Auto[,3:6])

```
describe(Auto[,3:6])
             vars   n    mean     sd median trimmed    mad  min    max  range
skew kurtosis    se
displacement    1 392  194.41 104.64  151.0  183.83  90.44   68  455.0  387.0
0.70    -0.79  5.29
horsepower      2 392  104.47  38.49   93.5   99.82  28.91   46  230.0  184.0
1.08     0.65  1.94
weight          3 392 2977.58 849.40 2803.5 2916.94 948.12 1613 5140.0 3527.0
0.52    -0.83 42.90
acceleration    4 392   15.54   2.76   15.5   15.48   2.52    8   24.8   16.8
0.29     0.41  0.14
```

The describe function provides the mean and standard deviation of each quantitative predictor.

4.(e) Remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

View(Auto)

AutoTemp <- Auto

View(AutoTemp)

AutoTemp <- AutoTemp[-(10:84),]

describe(AutoTemp)

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
mpg	1	317	24.37	7.88	23.9	23.97	9.04	11.0	46.6	35.6	0.40	-0.68	0.44
cylinders*	2	317	3.15	1.30	2.0	3.07	0.00	1.0	5.0	4.0	0.33	-1.63	0.07
displacement	3	317	187.75	99.94	146.0	176.80	83.03	68.0	455.0	387.0	0.80	-0.54	5.61
horsepower	4	317	100.96	35.90	90.0	96.84	29.65	46.0	230.0	184.0	1.18	1.25	2.02
weight	5	317	2939.64	812.65	2795.0	2879.29	941.45	1649.0	4997.0	3348.0	0.53	-0.72	45.64
acceleration	6	317	15.72	2.69	15.5	15.65	2.22	8.5	24.8	16.3	0.34	0.44	0.15
year*	7	317	8.13	3.11	8.0	8.15	4.45	1.0	13.0	12.0	-0.14	-0.86	0.17
origin*	8	317	1.60	0.82	1.0	1.50	0.00	1.0	3.0	2.0	0.85	-0.98	0.05
name*	9	317	148.37	88.90	148.0	147.71	117.13	1.0	304.0	303.0	0.05	-1.24	4.99

Mean, SD, Range after removing 75 observations

describe(Auto)

4.(f) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

Univariate Analysis of predictor varibles

#mpg

describe(Auto$mpg)

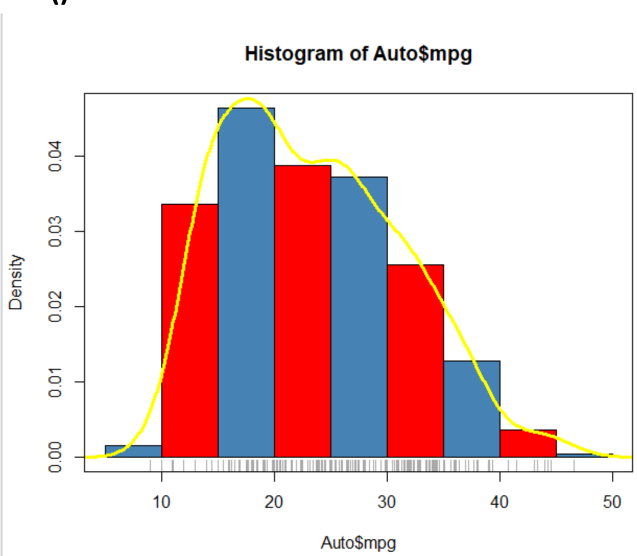
```
describe(Auto$mpg)
   vars   n  mean   sd median trimmed mad min  max range skew kurtosis   se
X1    1 392 23.45 7.81  22.75   22.99 8.6   9 46.6  37.6 0.45    -0.54 0.39
```

hist(Auto$mpg,col=c("steelblue", "red"), freq=F)

rug(jitter(Auto$mpg), col="darkgray")

lines(density(Auto$mpg), col="yellow", lwd=3)

box()



right skewed

#cylinders

t1<-table(Auto$cylinders)

t1

```
3   4   5   6   8
4 199   3  83 103
```

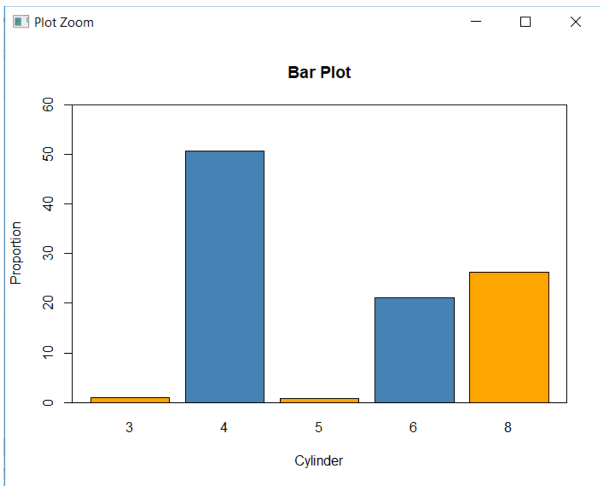
barplot(t1, main = "Bar Plot", xlab = "Cylinder", ylab = "Frequency")

pts1<-prop.table(t1)

pts1<-pts1*100 # Convert to percentages

barplot(pts1, main = "Bar Plot", xlab = "Cylinder", ylab = "Proportion", col=c("orange", "steelblue"), ylim=c(0,60))

box()



cylinder ratio is given as 4 cylinders > 8 cylinders > 6 cylinders. Whereas 3 & 5 cylinders are very low.

#displacement

describe(Auto$displacement)

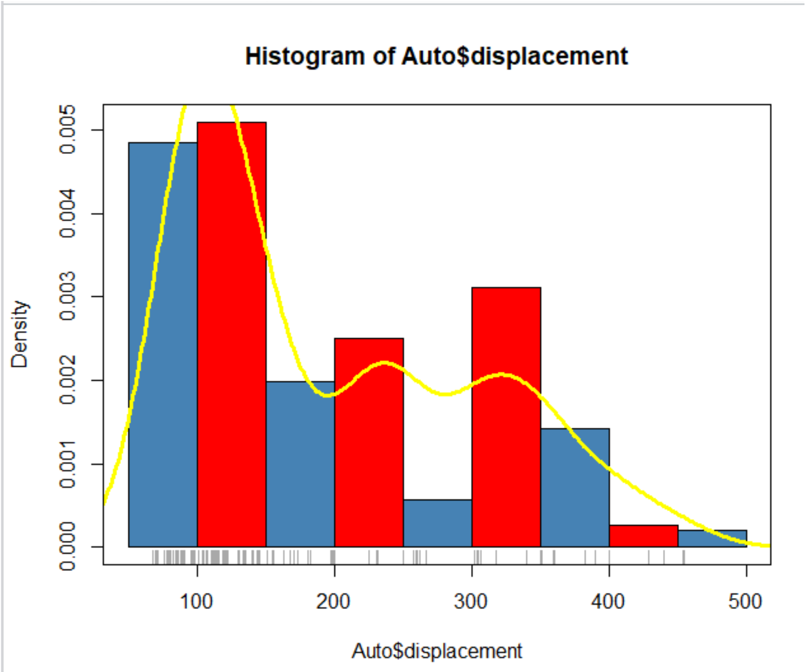
```
vars   n   mean     sd median trimmed   mad min max range skew kurtosis   se
X1    1 392 194.41 104.64    151  183.83 90.44  68 455   387  0.7    -0.79
5.29
```

hist(Auto$displacement,col=c("steelblue", "red"), freq=F)

rug(jitter(Auto$displacement), col="darkgray")

lines(density(Auto$displacement), col="yellow", lwd=3)

box()



right skewed

#horsepower

describe(Auto$horsepower)

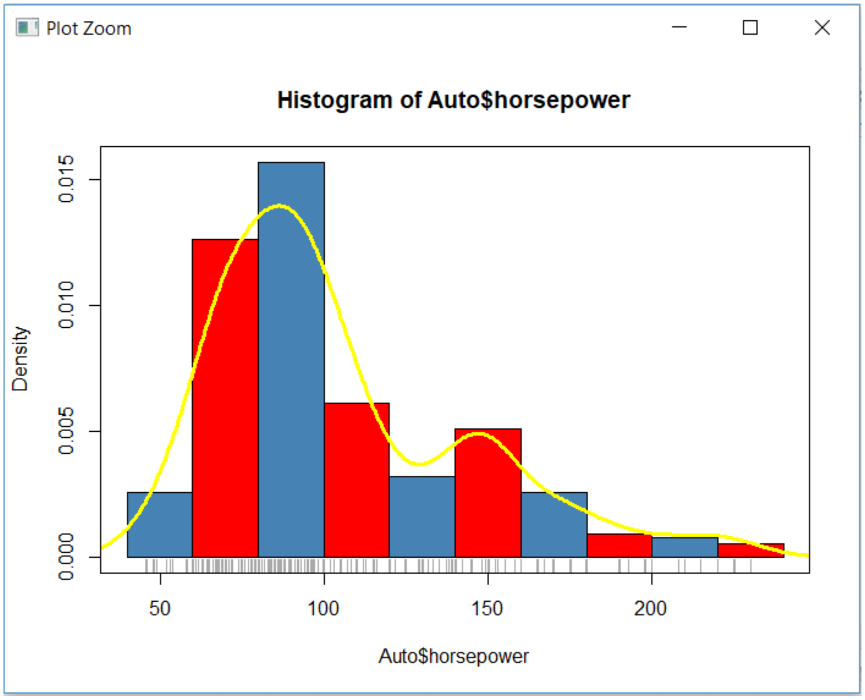
```
vars   n   mean    sd median trimmed   mad min max range skew kurtosis   se
X1    1 392 104.47 38.49   93.5   99.82 28.91  46 230   184 1.08     0.65
1.94
```

hist(Auto$horsepower,col=c("steelblue", "red"), freq=F)

rug(jitter(Auto$horsepower), col="darkgray")

lines(density(Auto$horsepower), col="yellow", lwd=3)

box()



right skewed

#weight

describe(Auto$weight)

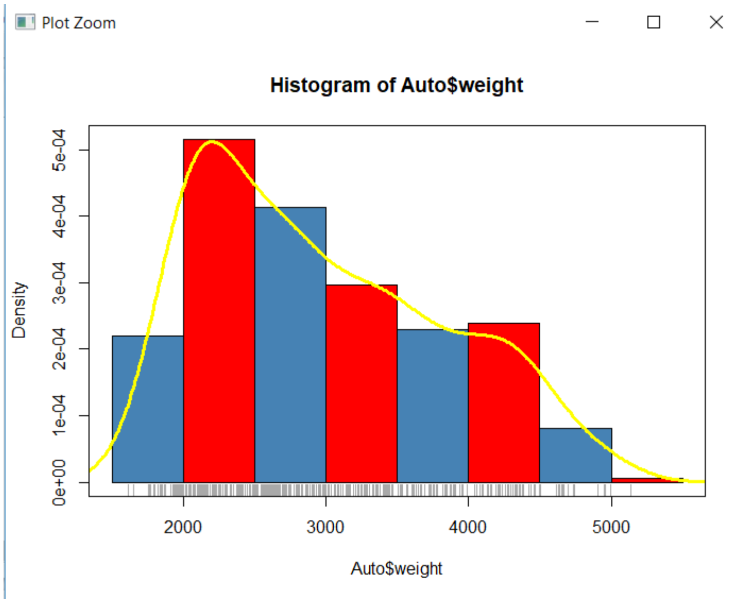
```
vars    n     mean     sd median trimmed    mad  min  max range skew kurtosis
se
X1     1 392 2977.58 849.4 2803.5 2916.94 948.12 1613 5140  3527 0.52    -0.83
42.9
```

hist(Auto$weight,col=c("steelblue", "red"), freq=F)

rug(jitter(Auto$weight), col="darkgray")

lines(density(Auto$weight), col="yellow", lwd=3)

box()



Right skewed

#acceleration

describe(Auto$acceleration)

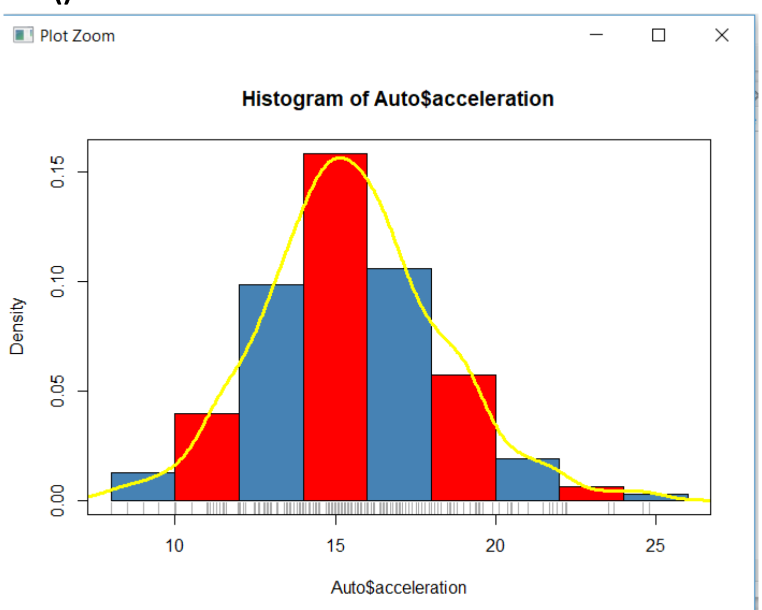
```
vars   n  mean   sd median trimmed  mad min  max range skew kurtosis   se
X1    1 392 15.54 2.76   15.5   15.48 2.52   8 24.8  16.8 0.29     0.41 0.14
```

hist(Auto$acceleration,col=c("steelblue", "red"), freq=F)

rug(jitter(Auto$acceleration), col="darkgray")

lines(density(Auto$acceleration), col="yellow", lwd=3)

box()



Normal distribution

#year

t1<-table(Auto$year)

t1

```
70 71 72 73 74 75 76 77 78 79 80 81 82
29 27 28 40 26 30 34 28 36 29 27 28 30
```

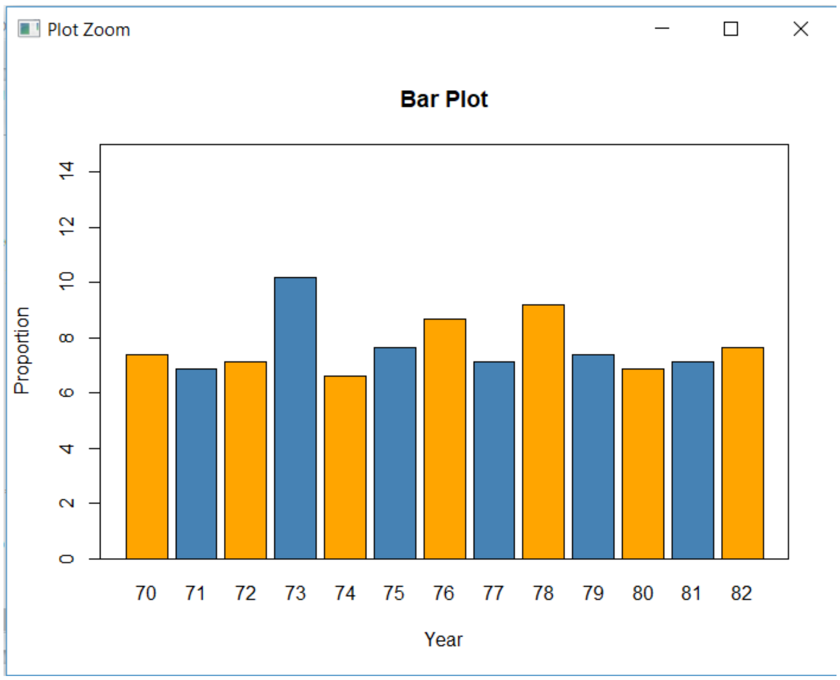
barplot(t1, main = "Bar Plot", xlab = "Year", ylab = "Frequency")

pts1<-prop.table(t1)

pts1<-pts1*100 # Convert to percentages

barplot(pts1, main = "Bar Plot", xlab = "Year", ylab = "Proportion", col=c("orange", "steelblue"), ylim=c(0,15))

box()



year ratio is almost evenly distributed between 7 to 9. only 73 year > 78 year > 76 year

#origin

t1<-table(Auto$origin)

t1

```
  1   2   3
245  68  79
```

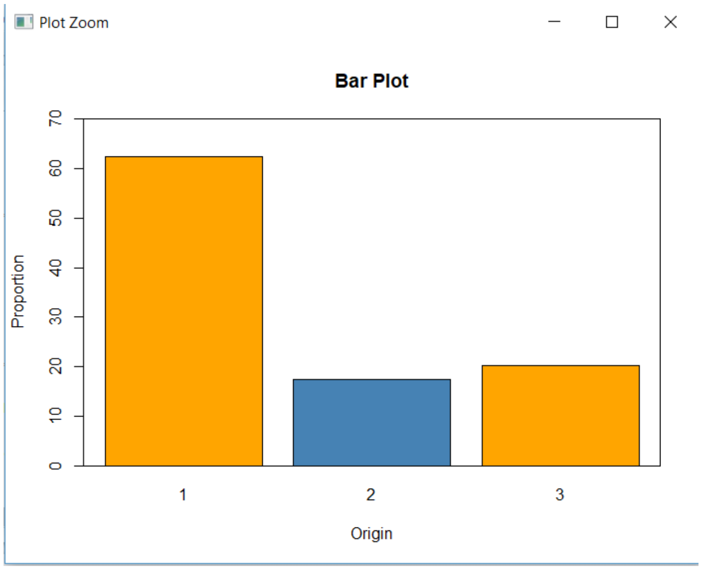
barplot(t1, main = "Bar Plot", xlab = "Origin", ylab = "Frequency")

pts1<-prop.table(t1)

pts1<-pts1*100 # Convert to percentages

barplot(pts1, main = "Bar Plot", xlab = "Origin", ylab = "Proportion", col=c("orange", "steelblue"), ylim=c(0,70))

box()



origin ratio is higher for 1 compared to 2 and 3 which are almost same.

#name

t1<-table(Auto$name)

t1

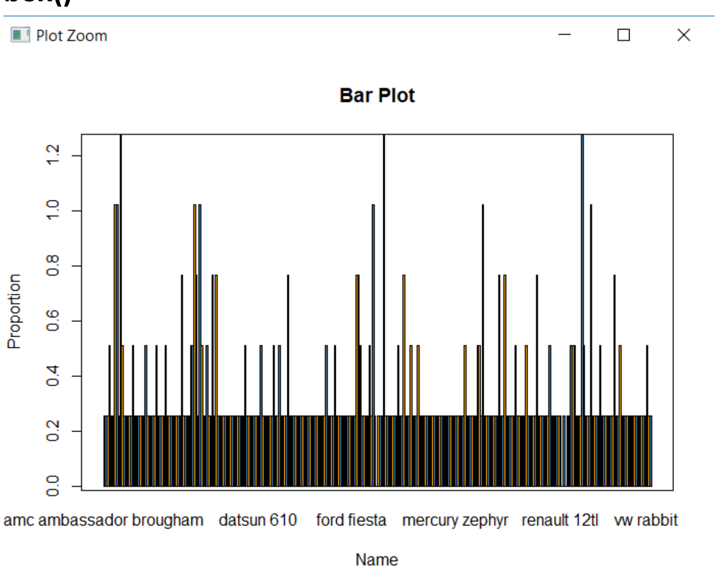
barplot(t1, main = "Bar Plot", xlab = "Name", ylab = "Frequency")

pts1<-prop.table(t1)

pts1<-pts1*100 # Convert to percentages

barplot(pts1, main = "Bar Plot", xlab = "Name", ylab = "Proportion", col=c("orange", "steelblue"))

box()



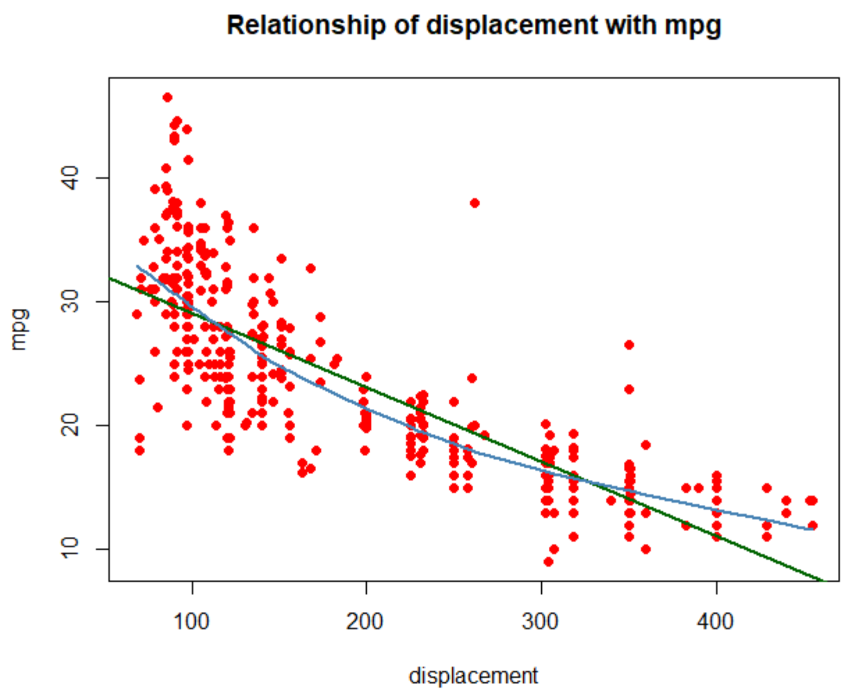
All cars have minimum value of 1 names. so 1 name > 2 names > 3 names > 4 names > 5 names

Bivariate Analysis

Relationship between numeric variables and mpg (Target Variable)

Relation between mpg and displacement

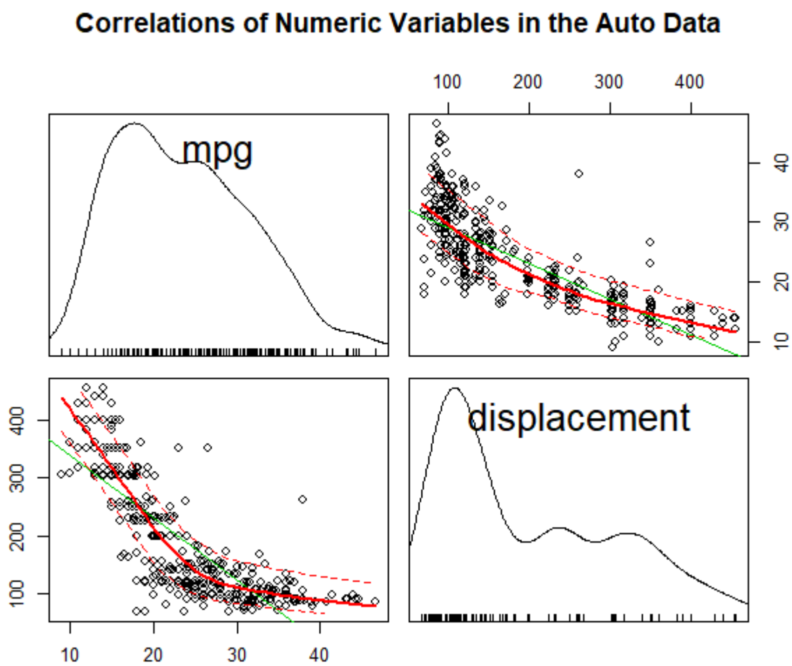
```
plot(Auto$mpg~Auto$displacement, col="red",
   main="Relationship of displacement with mpg",
   xlab="displacement",
   ylab="mpg",
   pch=16)

abline(lm(Auto$mpg~Auto$displacement), col="darkgreen", lwd=2.5)
lines(lowess(Auto$mpg~Auto$displacement), col="steelblue", lwd=2.5)
```



There is a negative relation between displacement and mpg As displacement increases the mpg decreases.

scatterplotMatrix(~mpg+displacement, data=Auto, main="Correlations of Numeric Variables in the Auto Data")

Doing a Scatterplot



Autonum <- Auto[,c(1,3)] # Making a numeric dataset

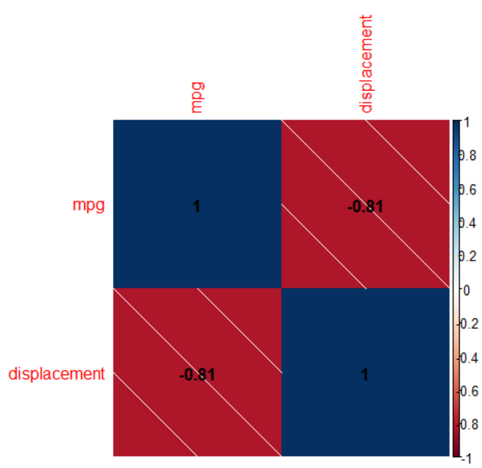
View(Autonum)

cormat <- cor(Autonum) # Correlation matrix

round(cormat, 2) # Rounding off to two decimal places

```
             mpg displacement
mpg         1.00        -0.81
displacement -0.81        1.00
```

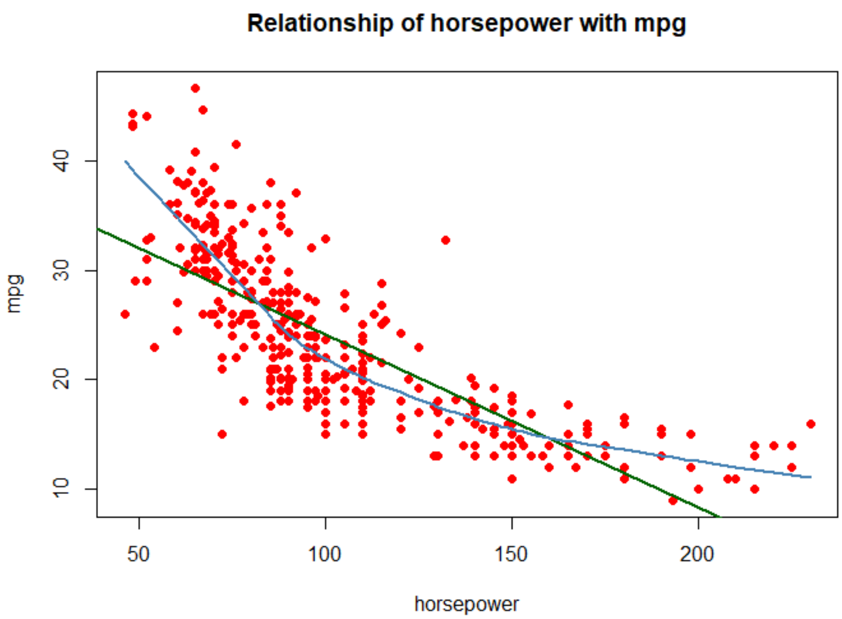
corrplot(cormat, method="shade", addCoef.col="black")



Making a corrplot to find various correlations

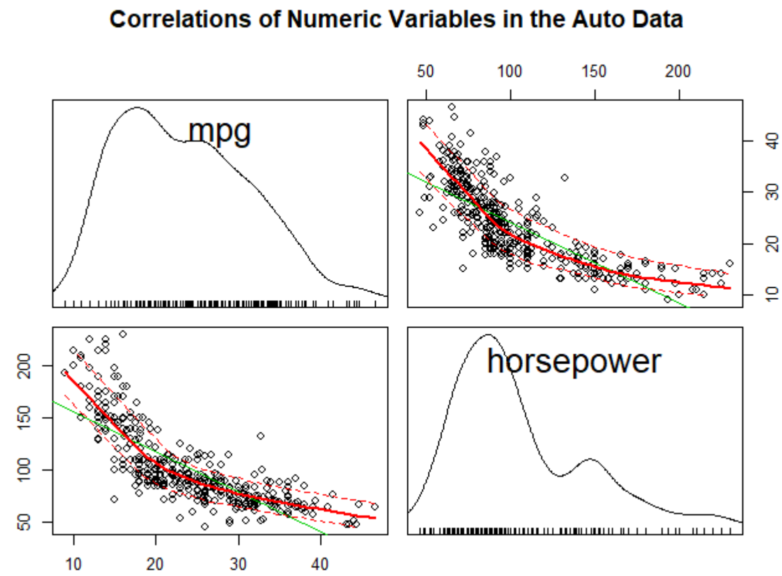
As observed there is a high negative correaltion between displacement and mpg and the corr value is -0.81

```
# Relation between mpg and horsepower
plot(Auto$mpg~Auto$horsepower, col="red",
    main="Relationship of horsepower with mpg",
    xlab="horsepower",
    ylab="mpg",
    pch=16)

abline(lm(Auto$mpg~Auto$horsepower), col="darkgreen", lwd=2.5)

lines(lowess(Auto$mpg~Auto$horsepower), col="steelblue", lwd=2.5)
```



There is a negative relation between horsepower and mpg As horsepower increases the mpg decreases.

```
scatterplotMatrix(~mpg+horsepower, data=Auto, main="Correlations of Numeric Variables in
the Auto Data") #Doing a Scatterplot
```



Autonum <- Auto[,c(1,4)] # Making a numeric dataset

View(Autonum)

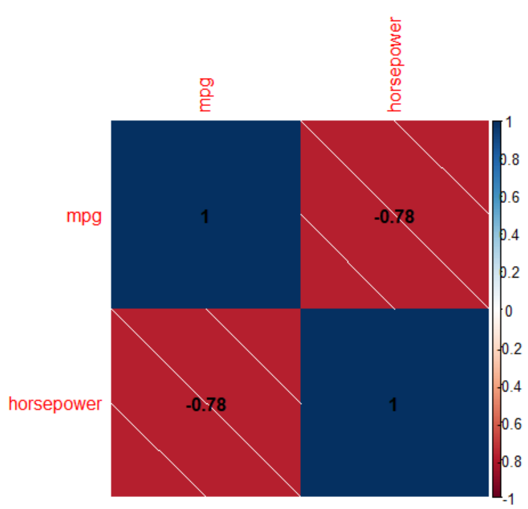
cormat <- cor(Autonum) # Correlation matrix

round(cormat, 2) # Rounding off to two decimal places

```
mpg horsepower
mpg              1.00       -0.78
horsepower -0.78        1.00
```

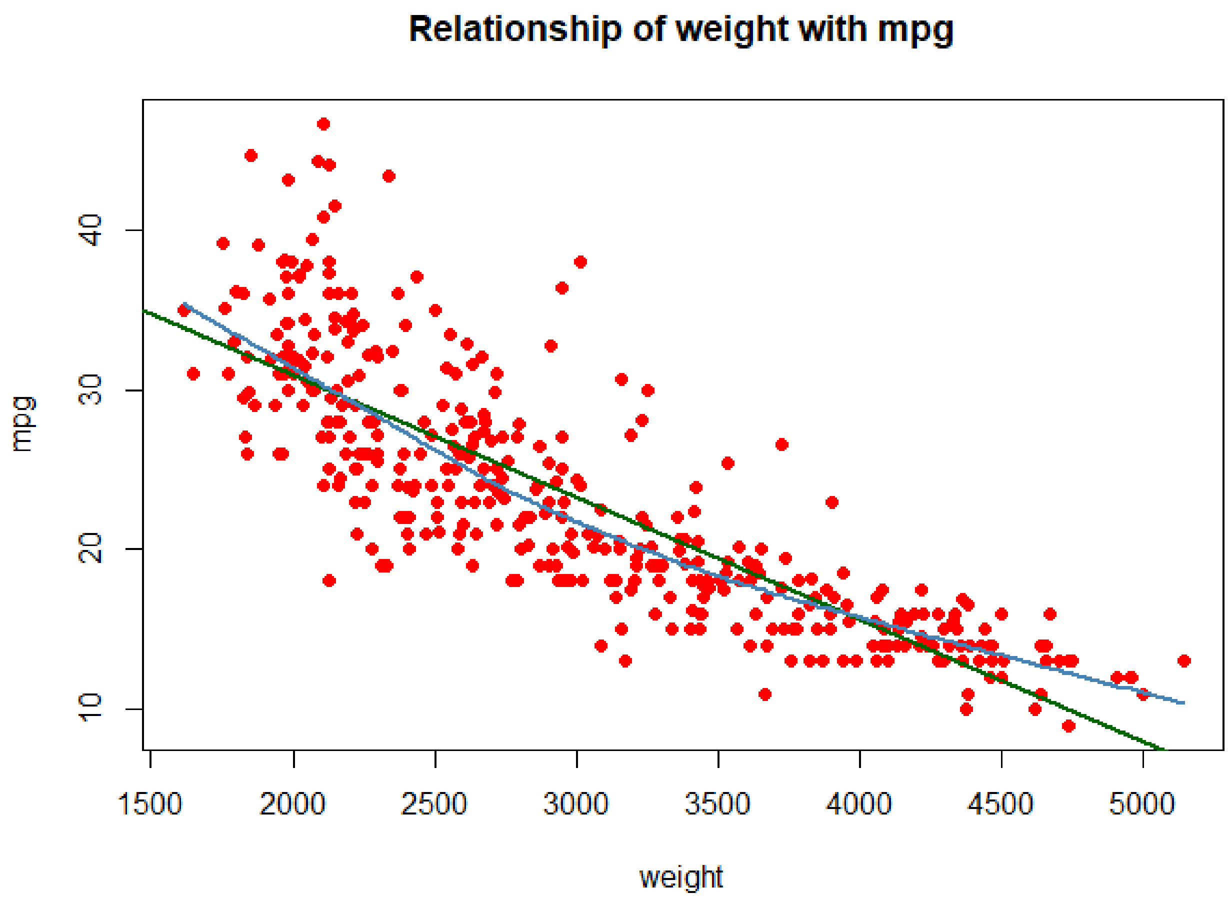
corrplot(cormat, method="shade", addCoef.col="black")

Making a corrplot to find various correlations



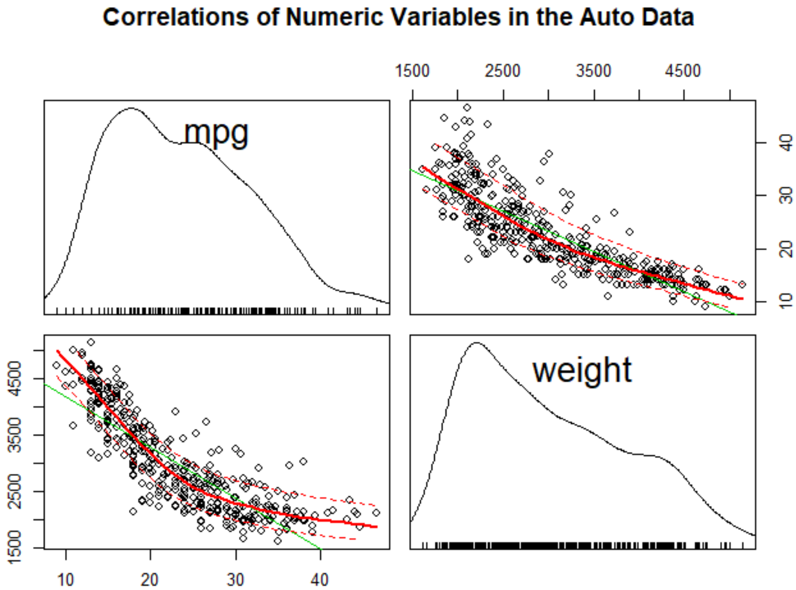
As observed there is a high negative correaltion between horsepower and mpg and the corr value is -0.78

```
# Relation between mpg and weight
plot(Auto$mpg~Auto$weight, col="red",
    main="Relationship of weight with mpg",
    xlab="weight",
    ylab="mpg",
    pch=16)

abline(lm(Auto$mpg~Auto$weight), col="darkgreen", lwd=2.5)

lines(lowess(Auto$mpg~Auto$weight), col="steelblue", lwd=2.5)
```



 # There is a negative relation between weight and mpg As weight increases the mpg decreases.

```
scatterplotMatrix(~mpg+weight, data=Auto, main="Correlations of Numeric Variables in the Auto Data") #Doing a Scatterplot
```



Autonum <- Auto[,c(1,5)] # Making a numeric dataset

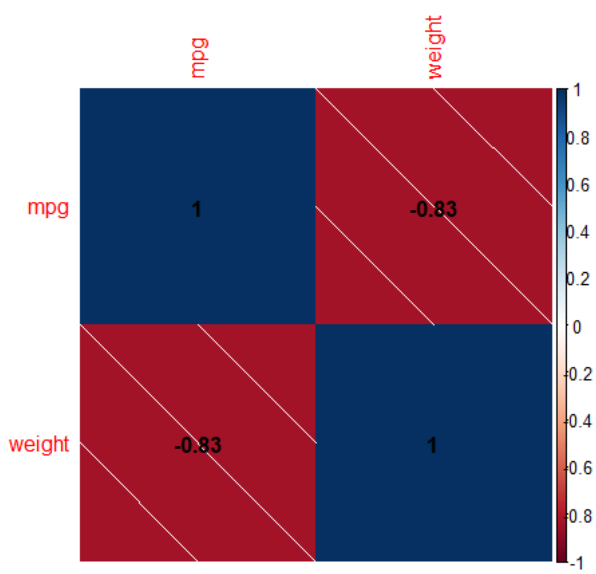
View(Autonum)

cormat <- cor(Autonum) # Correlation matrix

round(cormat, 2) # Rounding off to two decimal places

```
  mpg weight
mpg      1.00  -0.83
weight -0.83   1.00
```

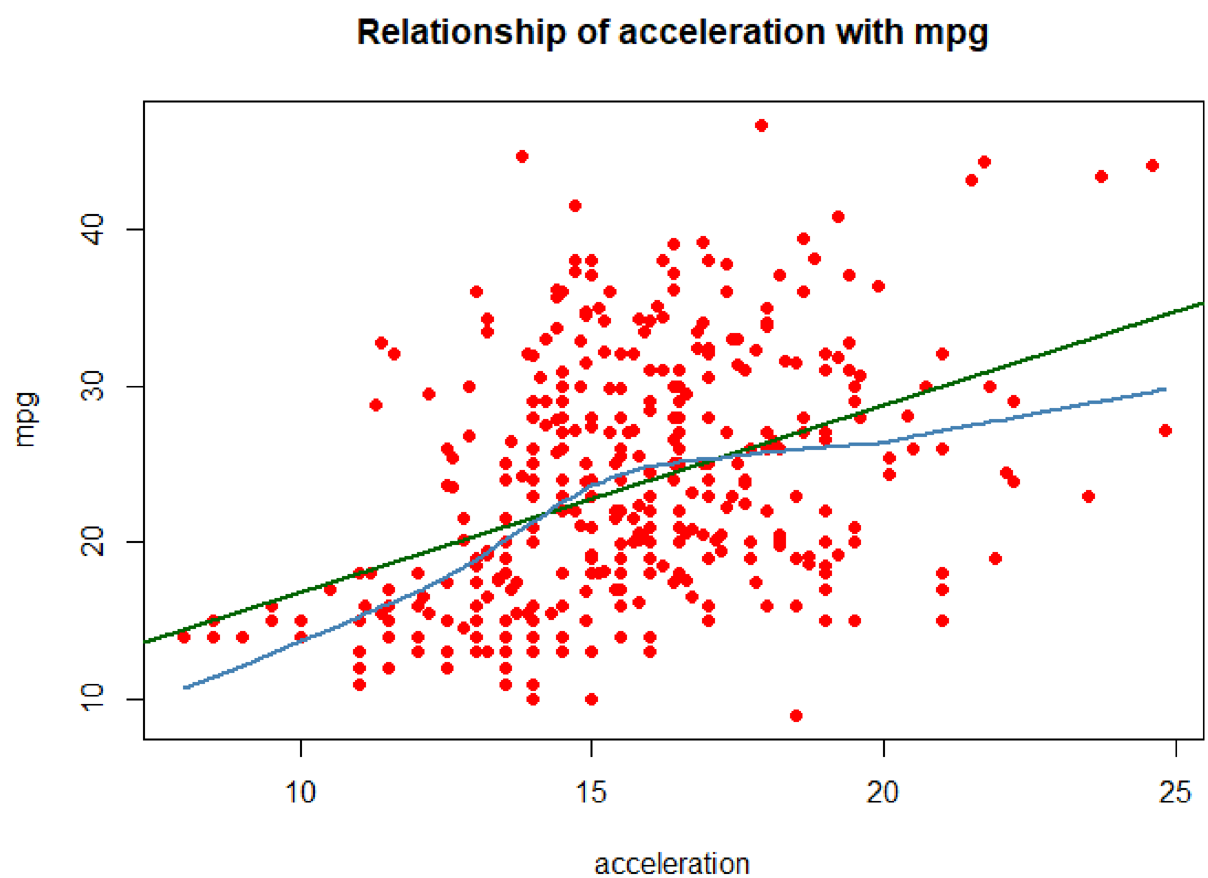
corrplot(cormat, method="shade", addCoef.col="black")



Making a corrplot to find various correlations

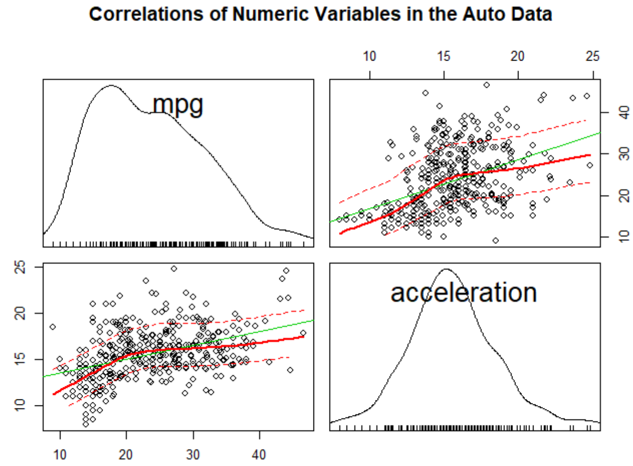
As observed there is a high negative correaltion between weight and mpg and the corr value is -0.83

```
# Relation between mpg and acceleration
plot(Auto$mpg~Auto$acceleration, col="red",
    main="Relationship of acceleration with mpg",
    xlab="acceleration",
    ylab="mpg",
    pch=16)

abline(lm(Auto$mpg~Auto$acceleration), col="darkgreen", lwd=2.5)

lines(lowess(Auto$mpg~Auto$acceleration), col="steelblue", lwd=2.5)
```



There is a positive relation between acceleration and mpg As acceleration increases the mpg increases and once again with increase in acceleraion the mpg seems to decrease, as we know that if we speed up the car, it burns out the fuel more.

```
scatterplotMatrix(~mpg+acceleration, data=Auto, main="Correlations of Numeric Variables in the Auto Data") #Doing a Scatterplot
```



Autonum <- Auto[,c(1,6)] # Making a numeric dataset

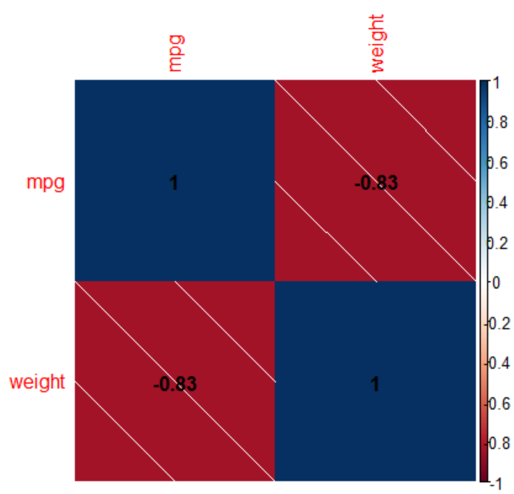
View(Autonum)

cormat <- cor(Autonum) # Correlation matrix

round(cormat, 2) # Rounding off to two decimal places

```
mpg weight
mpg      1.00  -0.83
weight -0.83   1.00
```

corrplot(cormat, method="shade", addCoef.col="black") # Making a corrplot to find various correlations



As observed there is a positive correaltion between acceleration and mpg and the corr value is 0.42

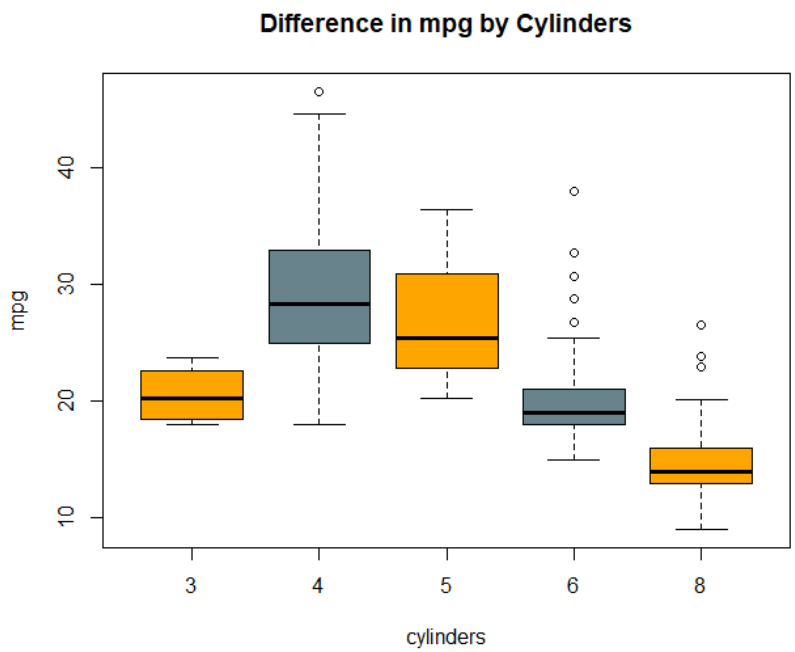
#Relationship between mpg and factor variables

#mpg and cylinders
describeBy(Auto$mpg , Auto$cylinders)

```
 Descriptive statistics by group
group: 3
   vars n  mean   sd median trimmed  mad min  max range skew kurtosis   se
X1    1 4 20.55 2.56  20.25   20.55 2.59  18 23.7   5.7 0.18    -2.15 1.28
------------------------------------------------------------------------------
group: 4
   vars   n  mean   sd median trimmed  mad min  max range skew kurtosis  se
X1    1 199 29.28 5.67   28.4      29 5.49  18 46.6  28.6 0.52        0 0.4
------------------------------------------------------------------------------
group: 5
   vars n  mean   sd median trimmed  mad  min  max range skew kurtosis   se
X1    1 3 27.37 8.23   25.4   27.37 7.56 20.3 36.4  16.1 0.23    -2.33 4.75
------------------------------------------------------------------------------
group: 6
   vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 83 19.97 3.83     19   19.44 2.22  15  38    23 2.06     5.99 0.42
------------------------------------------------------------------------------
group: 8
   vars   n  mean   sd median trimmed  mad min  max range skew kurtosis   se
X1    1 103 14.96 2.84     14   14.73 1.48   9 26.6  17.6 1.15     2.55 0.28
```

 # Mean and median across factor levels are almost same so it would be hard to find any relation.

 # BoxPlot
boxplot(mpg ~ cylinders, data=Auto, main="Difference in mpg by Cylinders",
xlab="cylinders", ylab="mpg",
col=c("orange", "lightblue4"))



 # As we can see that mpg is high for 4 cylinders, and the proportion goes by 4 > 5 > 3 > 6 > 8

#Relationship between mpg and factor variables

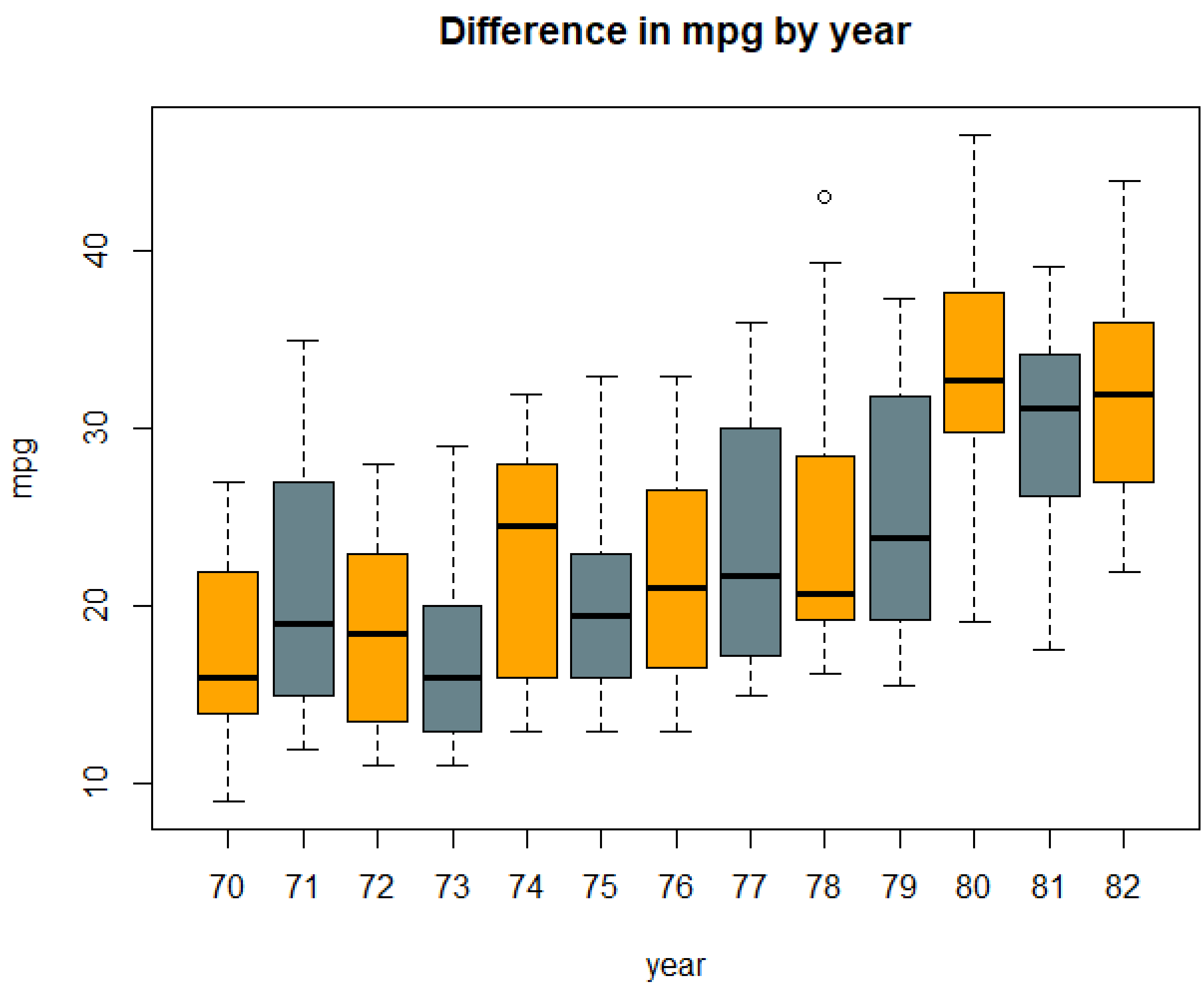
#mpg and year

describeBy(Auto$mpg , Auto$year)

 # Mean and median across factor levels are almost same so it would be hard to find any relation.

 # BoxPlot

```
boxplot(mpg ~ year, data=Auto, main="Difference in mpg by year",
    xlab="year", ylab="mpg",
    col=c("orange", "lightblue4"))
```



As we can see that mpg is increasing with increase in years. Highest for 80

#mpg and origin

describeBy(Auto$mpg , Auto$origin)

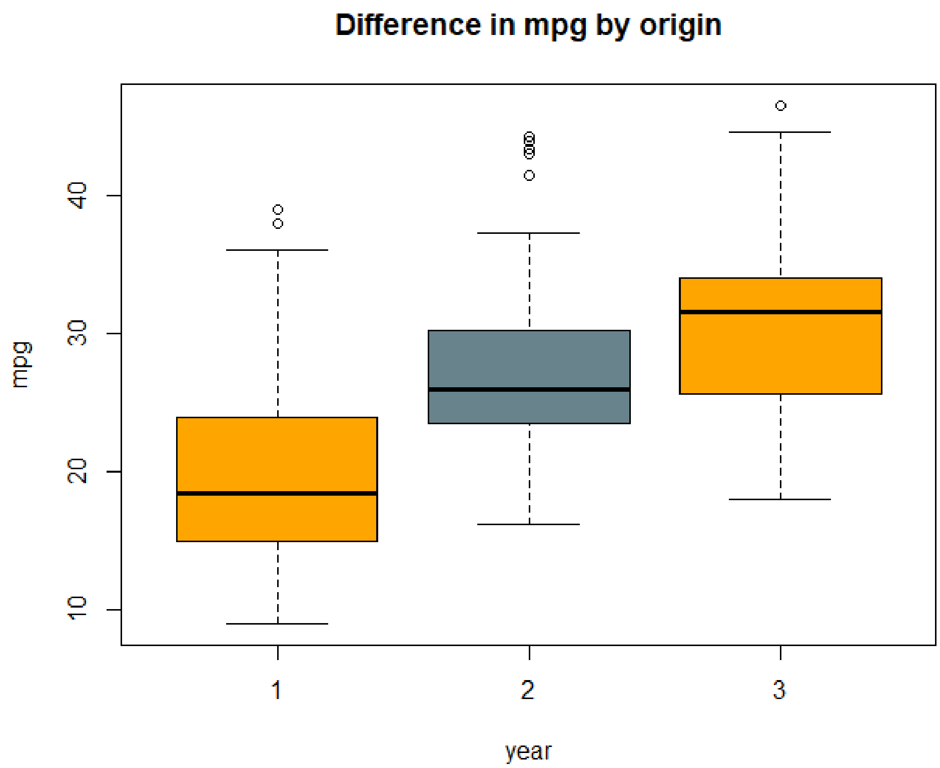
Mean and median across factor levels are almost same so it would be hard to find any relation.

```
 Descriptive statistics by group
group: 1
   vars   n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 245 20.03 6.44   18.5   19.37 6.67   9  39    30 0.83     0.03 0.41
--------------------------------------------------------------------------------
group: 2
   vars  n mean   sd median trimmed  mad  min  max range skew kurtosis  se
X1    1 68 27.6 6.58     26    27.1 5.78 16.2 44.3  28.1 0.73     0.31 0.8
--------------------------------------------------------------------------------
group: 3
   vars  n  mean   sd median trimmed  mad min  max range skew kurtosis   se
X1    1 79 30.45 6.09   31.6   30.47 6.52  18 46.6  28.6 0.01    -0.39 0.69
```

 # BoxPlot

boxplot(mpg ~ origin, data=Auto, main="Difference in mpg by origin",

```
xlab="year", ylab="mpg",
col=c("orange", "lightblue4"))
```

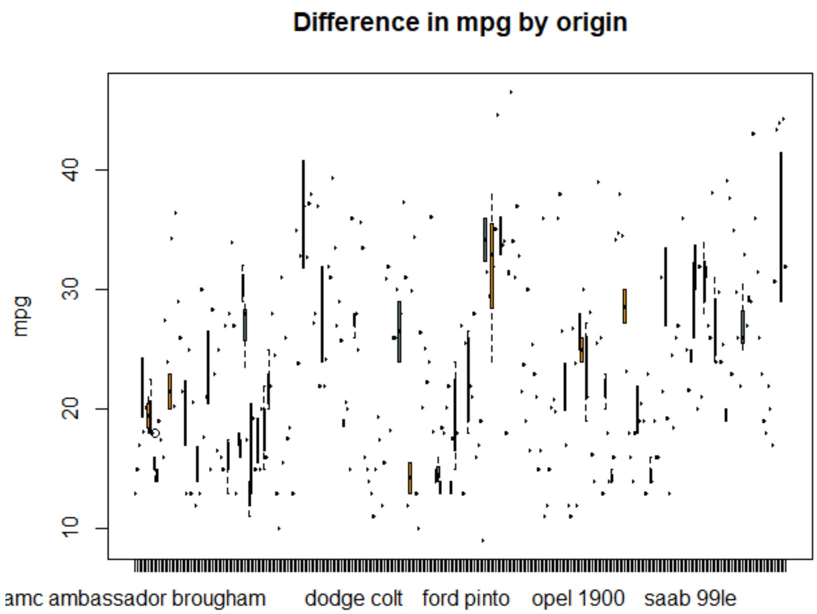


 # As we can see that mpg is high for 3 > 2 > 1

#mpg and name
describeBy(Auto$mpg , Auto$name)
Mean and median across factor levels are almost same so it would be hard to find any relation.

 # BoxPlot

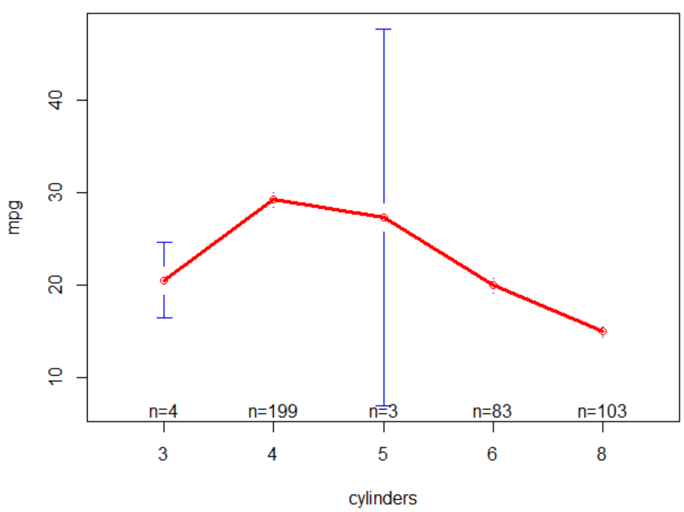
```
boxplot(mpg ~ name, data=Auto, main="Difference in mpg by name",
    xlab="name", ylab="mpg",
    col=c("orange", "lightblue4"))
```



 # Plot means

plotmeans(Auto$mpg~Auto$cylinders, xlab="cylinders", ylab="mpg", lwd=3, col="red")

We observe means of mpg to be differing for different cylinder capacities.



 # ANOVA

auto1.aov <- aov(mpg~cylinders, data=Auto)

auto1.aov

summary(auto1.aov)

```
auto1.aov
Call:
   aov(formula = mpg ~ cylinders, data = Auto)

Terms:
```

```
                cylinders Residuals
Sum of Squares  15274.507  8544.487
Deg. of Freedom         4       387

Residual standard error: 4.698806
Estimated effects may be unbalanced
> summary(auto1.aov)
             Df Sum Sq Mean Sq F value Pr(>F)
cylinders     4  15275    3819     173 <2e-16 ***
Residuals   387   8544      22
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The probability is very less so that we rject the null and accept that the means are different

 # We use Tukey pairwise comparisons

auto1.tk<-TukeyHSD(auto1.aov)

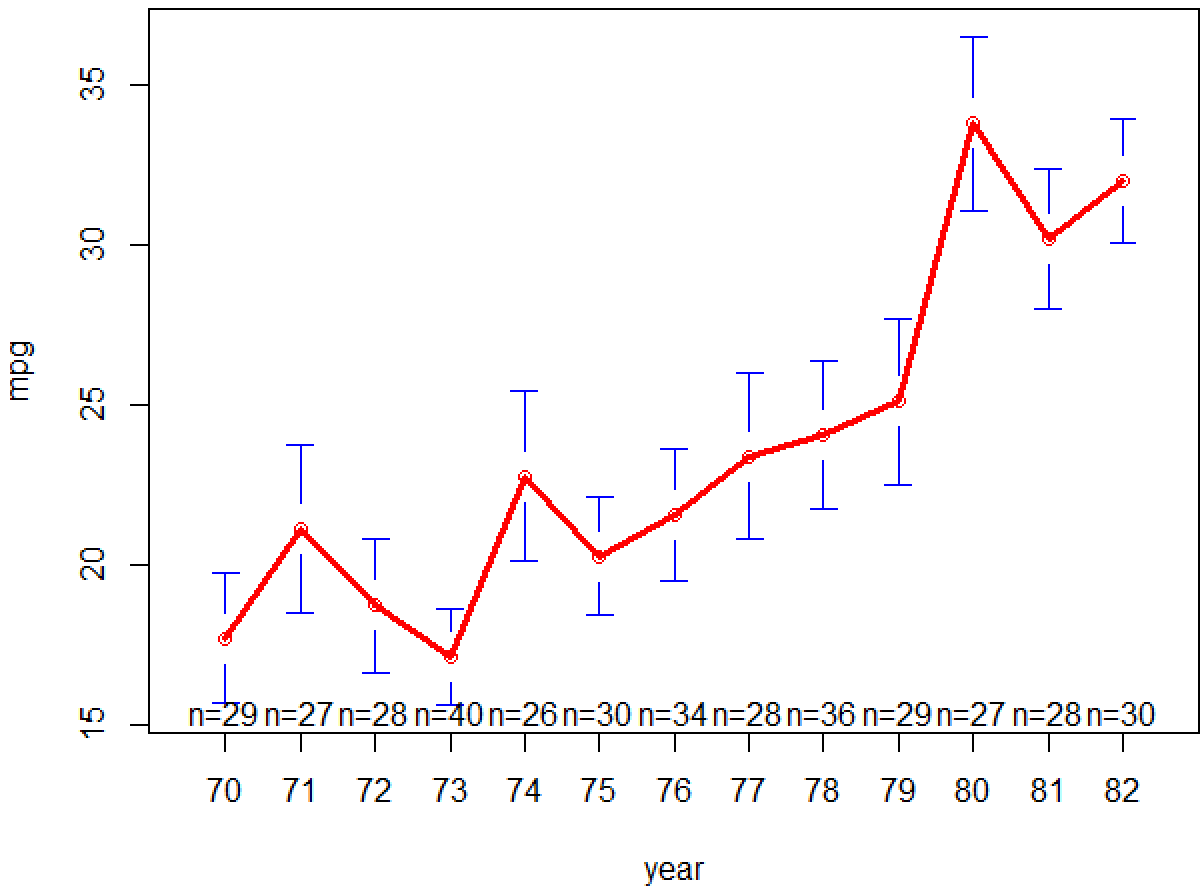
auto1.tk

```
Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = mpg ~ cylinders, data = Auto)

$cylinders
          diff        lwr         upr     p adj
4-3   8.733920   2.230560  15.2372794 0.0024534
5-3   6.816667  -3.019020  16.6523535 0.3193286
6-3  -0.576506  -7.168805   6.0157927 0.9992685
8-3  -5.586893 -12.149699   0.9759130 0.1366880
5-4  -1.917253  -9.408167   5.5736612 0.9560878
6-4  -9.310426 -10.993120  -7.6277312 0.0000000
8-4 -14.320813 -15.883977 -12.7576485 0.0000000
6-5  -7.393173 -14.961429   0.1750839 0.0592140
8-5 -12.403560 -19.946141  -4.8609787 0.0000851
8-6  -5.010387  -6.909913  -3.1108618 0.0000000
```

Thus there is an apparent relation between cylinders and mpg

 # Plot means

plotmeans(Auto$mpg~Auto$year, xlab="year", ylab="mpg", lwd=3, col="red")



We observe means of mpg to be differing for different years.

 # ANOVA

auto1.aov <- aov(mpg~year, data=Auto)

auto1.aov

summary(auto1.aov)

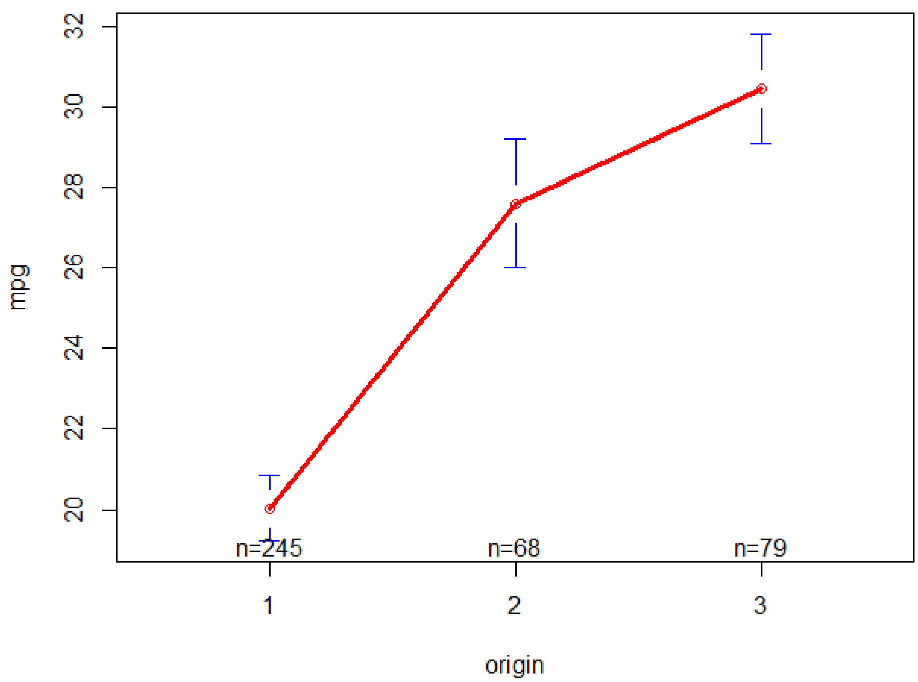
```
Call:
   aov(formula = mpg ~ year, data = Auto)

Terms:
                    year Residuals
Sum of Squares  10236.3   13582.7
Deg. of Freedom      12       379

Residual standard error: 5.986506
Estimated effects may be unbalanced
> summary(auto1.aov)
             Df Sum Sq Mean Sq F value Pr(>F)
year         12  10236   853.0    23.8 <2e-16 ***
Residuals   379  13583    35.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The probability is very less so that we rject the null and accept that the means are different

 # We use Tukey pairwise comparisons
auto1.tk<-TukeyHSD(auto1.aov)
auto1.tk

Most of the adjacent p values are not low enough to reject the nulls.
Thus there is no apparent relation between year and mpg

 # Plot means
plotmeans(Auto$mpg~Auto$origin, xlab="origin", ylab="mpg", lwd=3, col="red")



We observe means of mpg to be differing for different origins.

 # ANOVA
auto1.aov <- aov(mpg~origin, data=Auto)
auto1.aov
summary(auto1.aov)

```
Call:
   aov(formula = mpg ~ origin, data = Auto)

Terms:
                  origin Residuals
Sum of Squares   7904.291 15914.702
Deg. of Freedom         2       389

Residual standard error: 6.396236
Estimated effects may be unbalanced
```

```
> summary(auto1.aov)
             Df Sum Sq Mean Sq F value Pr(>F)    
origin        2   7904    3952    96.6 <2e-16 ***
Residuals   389  15915      41                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The probability is very less so that we rject the null and accept that the means are different

 # We use Tukey pairwise comparisons

auto1.tk<-TukeyHSD(auto1.aov)

auto1.tk

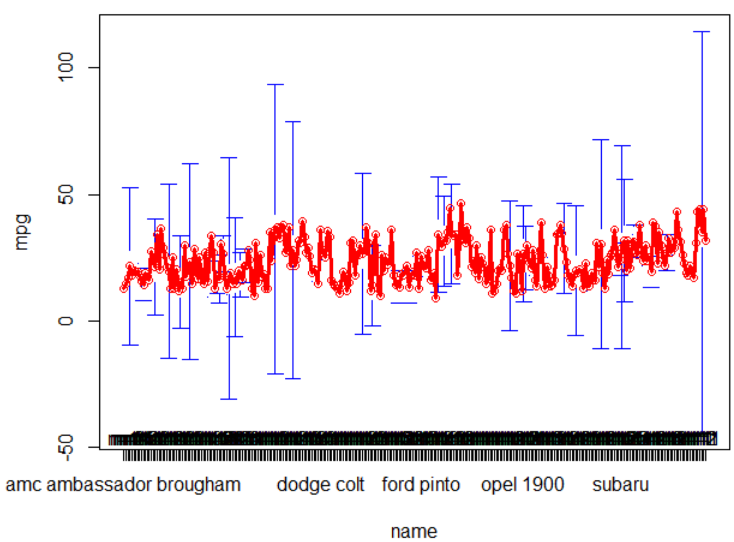
```
Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = mpg ~ origin, data = Auto)

$origin
         diff       lwr       upr     p adj
2-1  7.569472 5.5068042  9.632139 0.0000000
3-1 10.417164 8.4701431 12.364184 0.0000000
3-2  2.847692 0.3583458  5.337038 0.0202502
```

All the adjacent p values are low enough to reject the nulls.

Thus there is an apparent relation between origin and mpg

 # Plot means

plotmeans(Auto$mpg~Auto$name, xlab="name", ylab="mpg", lwd=3, col="red")



We observe means of mpg to be differing for different names.

 # ANOVA

auto1.aov <- aov(mpg~name, data=Auto)

auto1.aov

summary(auto1.aov)

```
Call:
   aov(formula = mpg ~ name, data = Auto)

Terms:
                    name Residuals
Sum of Squares  23039.24    779.75
Deg. of Freedom      300        91

Residual standard error: 2.92723
Estimated effects may be unbalanced
> summary(auto1.aov)
             Df Sum Sq Mean Sq F value Pr(>F)    
name        300  23039   76.80   8.963 <2e-16 ***
Residuals    91    780    8.57                   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The probability is very less so that we rject the null and accept that the means are different

 # We use Tukey pairwise comparisons

auto1.tk<-TukeyHSD(auto1.aov)

auto1.tk

Most of the adjacent p values are not low enough to reject the nulls.
Thus there is no apparent relation between year and name

4.(g) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

Yes, Of course. From the analysis of the plots that we obtained from the step (f) we are certain that the following variables might be useful for predicting mpg:
Numeric- Quantitative variables: displacement (-0.81), horsepower (-0.78), weight (-0.83) [correlation values]
Factor - Qualitative Variables : cylinders & origin [on the basis of anova & tukey pair tests]

Problem: 5

5(a) Import the data set.

salary_class <- read.csv(file.choose(), header = T)

View(salary_class) #32561 entries

str(salary_class)

```
'data.frame':   32561 obs. of  11 variables:
 $ AGE     : int  39 50 38 53 28 37 49 52 31 42 ...
 $ EMPLOYER: Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
 $ DEGREE  : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
 $ MSTATUS : Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
 $ JOBTYPE : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
 $ SEX     : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
 $ C.GAIN  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
 $ C.LOSS  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HOURS   : int  40 13 40 40 40 40 16 45 50 40 ...
 $ COUNTRY : Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 6 40 24 40 40 40 ...
 $ INCOME  : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

salary_class$EMPLOYER <- gsub("[^\\w\\s]", "", salary_class$EMPLOYER, perl=TRUE)

salary_class$JOBTYPE <- gsub("[^\\w\\s]", "", salary_class$JOBTYPE, perl=TRUE)

salary_class$COUNTRY <- gsub("[^\\w\\s]", "", salary_class$COUNTRY, perl=TRUE)

salary_class <- salary_class[salary_class$EMPLOYER != " ",]

salary_class <- salary_class[salary_class$JOBTYPE != " ",]

salary_class <- salary_class[salary_class$COUNTRY != " ",]

is.na(salary_class)

salary_class<-na.omit(salary_class) # 6 NAs are removed

salary_class$EMPLOYER <- as.factor(salary_class$EMPLOYER)

salary_class$JOBTYPE <- as.factor(salary_class$JOBTYPE)

salary_class$COUNTRY <- as.factor(salary_class$COUNTRY)

View(salary_class) #30162 entries

str(salary_class)

```
'data.frame':   30162 obs. of  11 variables:
 $ AGE     : int  39 50 38 53 28 37 49 52 31 42 ...
 $ EMPLOYER: Factor w/ 7 levels " Federalgov",..: 6 5 3 3 3 3 3 5 3 3 ...
 $ DEGREE  : Factor w/ 16 levels " 10th"," 11th",..: 10 10 12 2 10 13 7 12 13 10 ...
 $ MSTATUS : Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
 $ JOBTYPE : Factor w/ 14 levels " Admclerical",..: 1 4 6 6 10 4 8 4 10 4 ...
 $ SEX     : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
 $ C.GAIN  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
 $ C.LOSS  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HOURS   : int  40 13 40 40 40 40 16 45 50 40 ...
 $ COUNTRY : Factor w/ 41 levels " Cambodia"," Canada",..: 39 39 39 39 5 39 23 39 39 39 ...
 $ INCOME  : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

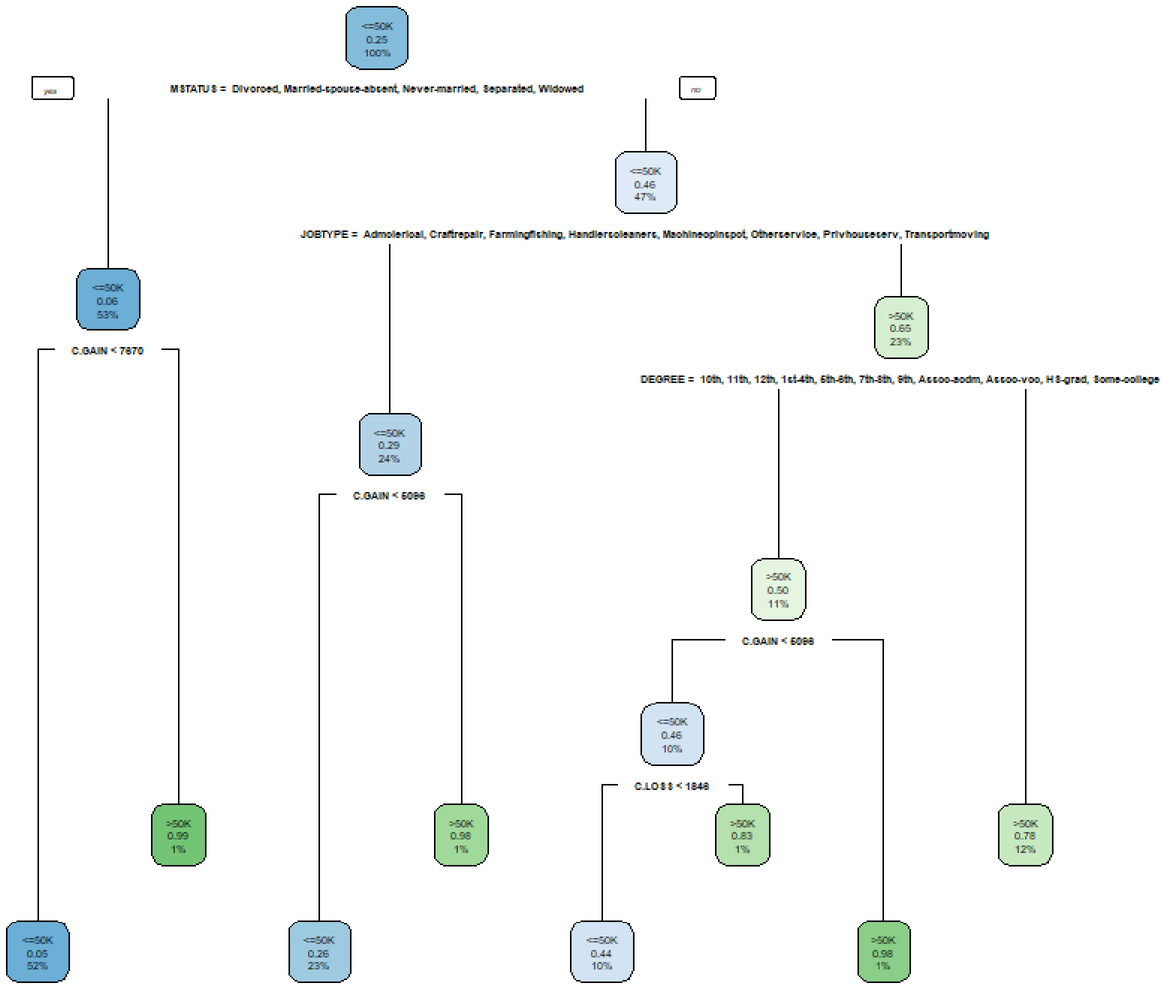
5.a)Use a partition node to divide the data into 60% train, 40% test.

```
set.seed(1234)

index = sample(2,nrow(salary_class), replace=TRUE, prob = c(0.6,0.4))
index

TrainData = salary_class[index == 1, ]
TrainData
nrow(TrainData)
```
17985

```
TestData = salary_class[index == 2, ]
TestData
nrow(TestData)
```
12177

5.(b) Create the default C&R decision tree. How many leaves are in the tree?

```
#RPART
install.packages("rpart")
library(rpart)

salary_class_rpart_default = rpart(INCOME~., data = TrainData)
salary_class_rpart_default

#Plotting the Tree
library("rpart.plot")
rpart.plot(salary_class_rpart_default)
```



As we can see from the plot that there are 8 leaves(Terminal nodes) in the Tree

5.(c) What are the major predictors of INCOME? Justify your choice. How can you get this information from the software?

summary(salary_class_rpart_default)

```
Call:
rpart(formula = INCOME ~ ., data = TrainData)
  n= 17985

          CP nsplit rel error    xerror       xstd
1 0.13194444      0 1.0000000 1.0000000 0.01283965
2 0.03968254      2 0.7361111 0.7548501 0.01160738
3 0.03461199      3 0.6964286 0.7206790 0.01140182
4 0.01444004      4 0.6618166 0.6593915 0.01100877
5 0.01000000      7 0.6164021 0.6571869 0.01099401
```

```
Variable importance
 MSTATUS  JOBTYPE   C.GAIN      SEX   DEGREE      AGE    HOURS  EMPLOYER
C.LOSS
      30       18       12       11        8        8        6        5
1

Node number 1: 17985 observations,    complexity param=0.1319444
  predicted class= <=50K  expected loss=0.2522102  P(node) =1
    class counts: 13449  4536
   probabilities: 0.748 0.252
  left son=2 (9551 obs) right son=3 (8434 obs)
  Primary splits:
      MSTATUS splits as  LRRLLLL, improve=1430.5260, (0 missing)
      C.GAIN  < 5095.5 to the left,  improve= 903.7522, (0 missing)
      DEGREE  splits as  LLLLLLLLLRRLRLRL, improve= 704.3209, (0 missing)
      JOBTYPE splits as  LLLRLLLLLRRLLL, improve= 637.0870, (0 missing)
      AGE     < 29.5   to the left,  improve= 564.5198, (0 missing)
  Surrogate splits:
      SEX      splits as  LR, agree=0.698, adj=0.355, (0 split)
      AGE      < 33.5   to the left,  agree=0.648, adj=0.250, (0 split)
      JOBTYPE  splits as  LLRRRLLLLRRLLR, agree=0.624, adj=0.199, (0 split)
      HOURS    < 43.5   to the left,  agree=0.599, adj=0.144, (0 split)
      EMPLOYER splits as  RLLRRRLR, agree=0.579, adj=0.101, (0 split)
```

From the summary function we get to know the major predictors of INCOME as given below:
Variable importance

MSTATUS	JOBTYPE	C.GAIN	SEX	DEGREE	AGE	HOURS	EMPLOYER	C.LOSS
30	18	12	11	8	8	6	5	1

MSTATUS JOBTYPE C.GAIN SEX DEGREE AGE HOURS EMPLOYER C.LOSS are the major predictors as they have high variable importance in the mentioned order. As we can see above, we can use both summary function and the rpart.plot function to find the major predictors of income. When we see the plot we can find that the first node gets split based on marital status, followed by Job Type and then c.gain, which is further split by Sex and degree.

From the Tree we can find the nodes which are green gives the outcomes of INCOME > 50k. But none of the 7 green outcomes having INCOME> 50k have confidence more than 75% and only 3 of the 7 outcomes have support more than 5%. The best outcome of INCOME >50k is the one having 11% support and 50% confidence.

All the blue outcomes are the INCOMES <= 50K.Out of 8 outcomes of INCONES <=50K only 2 outcomes satisfy the given condition having support more than 5% and confidence more than 90%. Remaining 6 outcomes satisfy only the support condition but not the confidence condition.

5.(d)

So from the above criteria of support and conditions, we can come up with the following three best Rules as given below:

1. When the Marital Status= [Divorced, Married-spouse-absent, Never-Married, Separated, Widowed] And C.GAIN < 7670 -> Then INCOME <=50K
Support = 52% confidence = 95 %

2. When Marital Status = [Married-AF-spouse, Married-civ-spouse And JOBTYPE = Execmanagerial, Profspecialty, Sales, Techsupport, Protectiveserv, ArmedForces] And DEGREE = [HS-grad, 11th, 9th, Some-college, Assoc-acdm, 7th-8th, Assoc-voc, 5th-6th, 10th, 12th] -> Then INCOME > 50K
Support = 11% confidence = 50 %

3. When MSTATUS= [Divorced, Married-spouse-absent, Never-married, Separated, Widowed] And JOBTYPE= [Admclerical, Craftrepair, Farmingfishing, Handlerscleaners, Machineopinspct, Otherservice, Privhouseserv, Transportmoving] And DEGREE= [10th, 11th, 12th, 1st-4th, 5th-6th, 7th-8th, 9th, Assoc-acdm, Assoc-voc, HS-grad, Some-college] And [C.GAIN< 5095.5] ->Then INCOME <=50K
Support = 10% confidence = 44 %

table(predict(salary_class_rpart_default, type = "class"), TrainData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 15.54%

```
Actual
Predicted  <=50K  >50K
    <=50K  12945  2292
    >50K     504  2244
```

table(predict(salary_class_rpart_default, type = "class", newdata = TestData), TestData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 15.2%

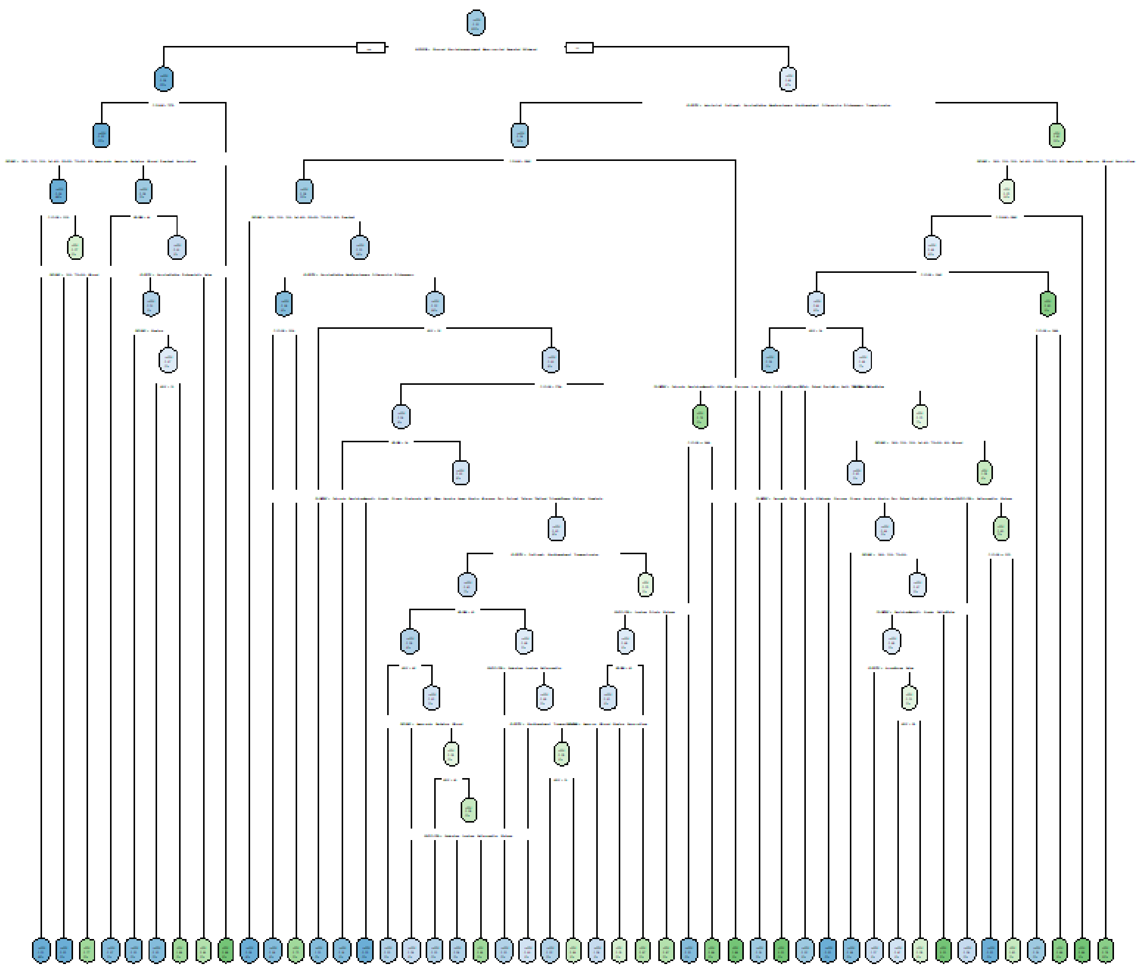
```
Actual
Predicted  <=50K  >50K
    <=50K   8837  1483
    >50K     368  1489
```

5.(e)Create two more C&R trees. The first is just like the default tree except you do not \prune tree to avoid overtting" (you need to let the model to grow to its full depth).

Creating one more default Tree without pruning and allowing the tree to grow fully salary_class_rpart_full_depth = rpart(INCOME~., data = TrainData,control = rpart.control(cp = 0.001))

salary_class_rpart_full_depth

rpart.plot(salary_class_rpart_full_depth)



summary(salary_class_rpart_full_depth)

Testing Accuracy on Training Data

table(predict(salary_class_rpart_full_depth, type = "class"), TrainData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 13.26%

```
          Actual
Predicted  <=50K  >50K
    <=50K  12544  1481
    >50K     905  3055
```

Testing Accuracy on Test Data

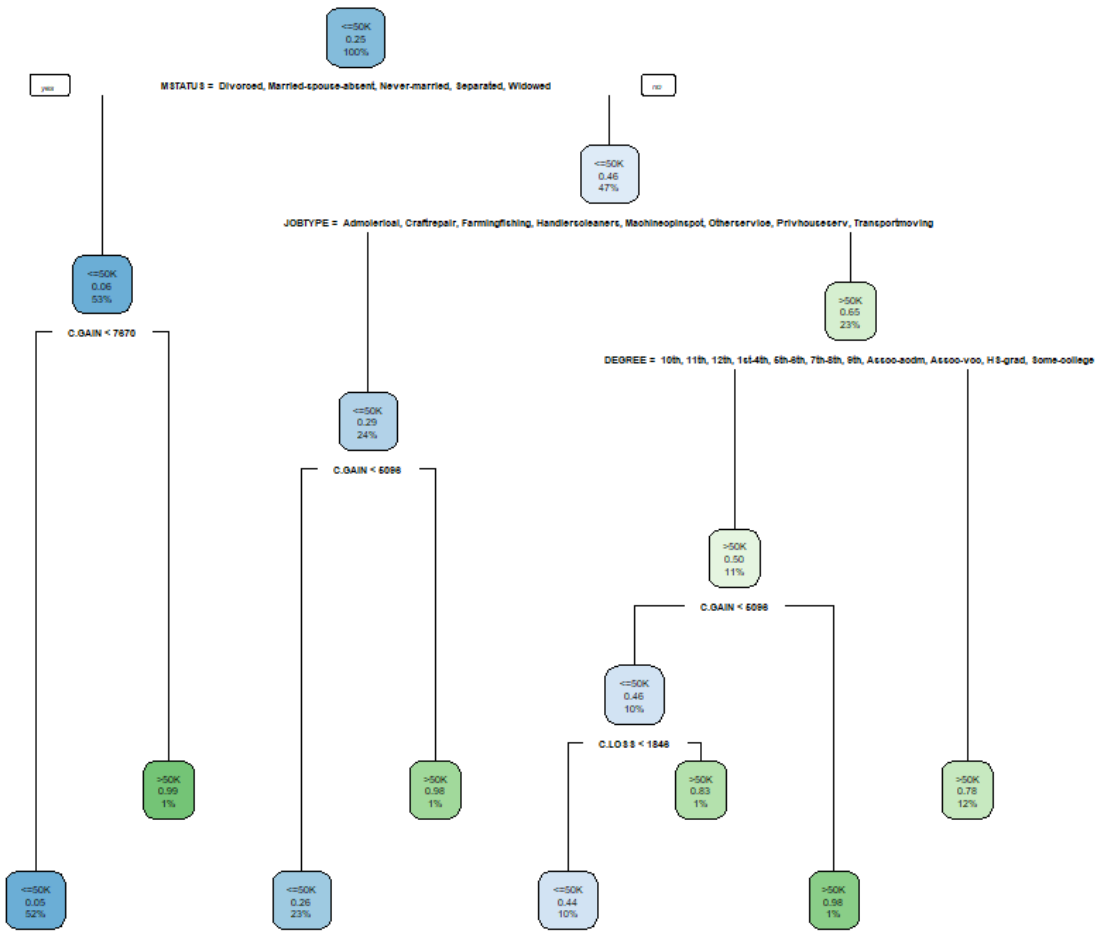
table(predict(salary_class_rpart_full_depth, type = "class", newdata = TestData), TestData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 14.58%

```
          Actual
Predicted  <=50K  >50K
```

```
<=50K     8469   1040
>50K       736   1932
```

The other does prune,but you require 500 records in a parent branch and 100 records in a child branch.

Creating a Pruned Tree with minsplit & minbucket parameters
salary_class_rpart_pruned = rpart(INCOME~., data = TrainData, control = rpart.control(minsplit = 500, minbucket = 100))
salary_class_rpart_pruned
rpart.plot(salary_class_rpart_pruned)



summary(salary_class_rpart_pruned)

#Testing Accuracy on Training Data
table(predict(salary_class_rpart_pruned, type = "class"), TrainData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 15.54%

```
         Actual
Predicted  <=50K  >50K
```

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

#Testing Accuracy on Test Data

table(predict(salary_class_rpart_full_depth, type = "class", newdata = TestData), TestData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 15.20%

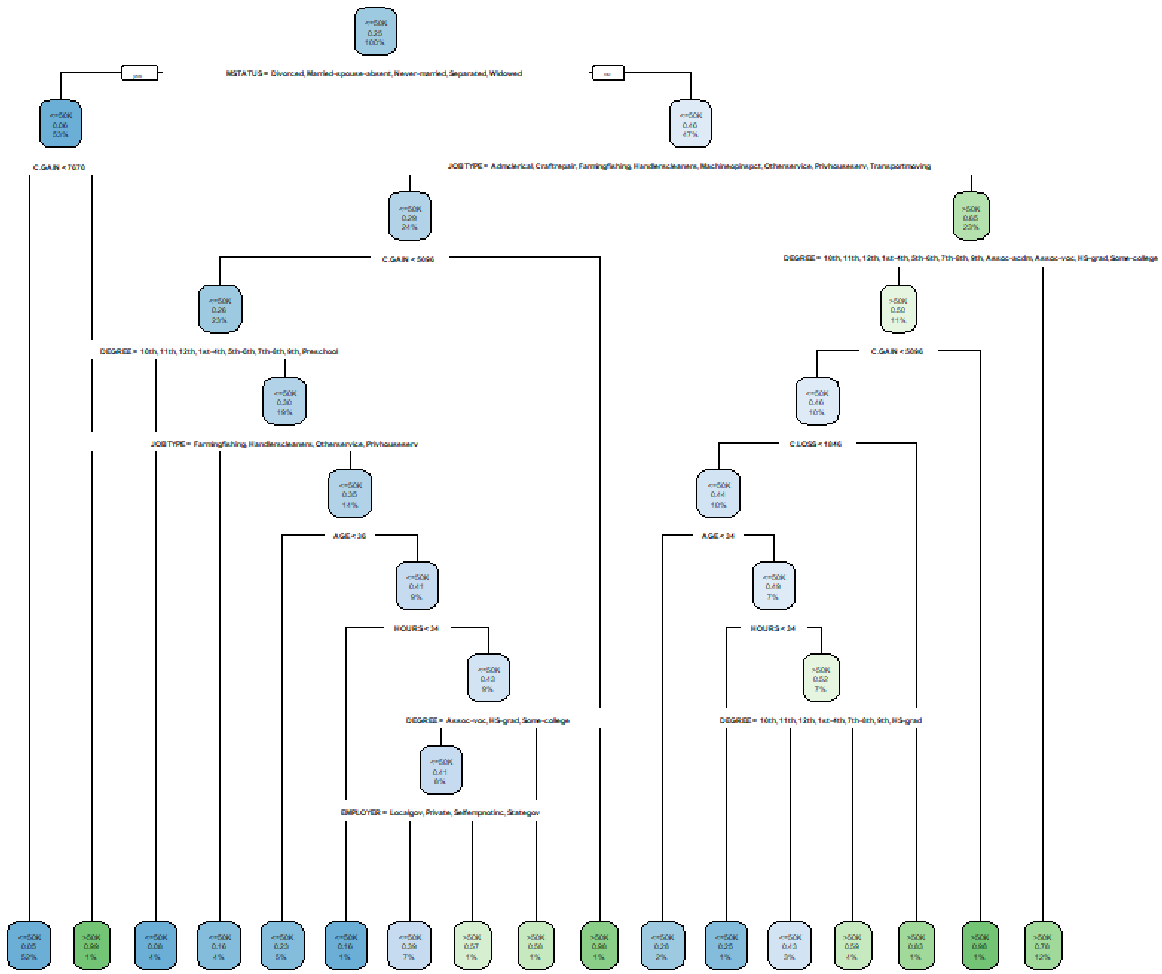
```
        Actual
Predicted  <=50K  >50K
    <=50K   8469  1040
    >50K     736  1932
```

We can observe that error rate of Training & Testing Data of Both Default Tree as we ll as Pruned Tree to be the same. So from this we can infer that the default one is actually pruning the tree.

Now when these two trees are compared to the Fully grown Tree we can find that the Fully grown tree performs better both in Training data as well as Test data as it has better Error rate in both Training as well as Test Data.

Creating a Pruned Tree with minsplit & minbucket parameters along with Tweaking Complexity parameter:

salary_class_rpart_full_pruned_cp = rpart(INCOME~., data = TrainData, control = rpart.control(minsplit = 500, minbucket = 100, cp = 0.001))
salary_class_rpart_full_depth
rpart.plot(salary_class_rpart_full_pruned_cp)



summary(salary_class_rpart_full_pruned_cp)

Testing Accuracy on Training Data

table(predict(salary_class_rpart_full_pruned_cp, type = "class"), TrainData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 14.61%

```
         Actual
Predicted  <=50K  >50K
    <=50K  12544  1724
    >50K     905  2812
```

Testing Accuracy on Test Data

table(predict(salary_class_rpart_full_pruned_cp, type = "class", newdata = TestData), TestData$INCOME, dnn = c("Predicted", "Actual")) #Error rate = 14.86%

```
         Actual
Predicted  <=50K  >50K
    <=50K   8529  1134
    >50K     676  1838
```

By adding complexity parameter to a pruned tree improves its error rate, but its error rate is still higher than the full depth tree.

Overall its the Full depth Tree which seems to be most accurate in both the Training Data as well as the Test Data because we are allowing the tree to grow to its full depth.