Meghashree Maddihally Nagoji

2/8/2022

Problem 1.

Explain what each line of the following R code do? You can run them in R and check the results.

1.a

```
x <-c(1,2.3,2,3,4,8,12,43,-4,-1)
x

##  [1]  1.0  2.3  2.0  3.0  4.0  8.0 12.0 43.0 -4.0 -1.0
```

Creates a vector and assigns it to variable x.

1.b

```
max(x)

## [1] 43
```

Max function takes the vector(x) as input parameter and outputs the maximum element in the vector x i.e. 43.

1.c

```
y<-c(x,NA)
y

##  [1]  1.0  2.3  2.0  3.0  4.0  8.0 12.0 43.0 -4.0 -1.0   NA
```

Creates a vector y with the elements of x and appends NA to the end of the vector.

1.d

```
max(y,na.rm = T)

## [1] 43
```

Prints the maximum element of vector y without considering NA. i.e. 43.

1.e

```
x2 <-c(-100,-43,0,3,1,-3)
min(x,x2)
```

```
## [1] -100
```

Creates a vector x2 and the min() function prints the minimum element of all the elements present in vector x and x2.

1.f

```
sample(4:10)

## [1] 10  9  4  7  8  5  6
```

The sample() function prints elements from 4 to 10 in random order.

1.g

```
sample(c(2,5,3),size = 3,replace = FALSE)

## [1] 3 5 2
```

A vector is created with elements 2,5 and 3. Three elements are selected at random from the created vector. The elements cannot be repeated.

1.h

```
sample(c(2,5,3), size = 3,replace = TRUE)

## [1] 2 5 5
```

A vector is created with elements 2,5 and 3. Three elements are selected at random from the created vector. The elements can be repeated.

1.i

```
sample(2,10, replace = TRUE)

##  [1] 1 1 2 2 2 1 1 2 2 1
```

Ten elements are selected from 1 to 2 at random. The values can be repeated.

1.j

```
sample(1:2,size=10,prob=c(1,3),replace=TRUE)

##  [1] 2 2 2 2 2 2 2 2 2 2
```

Selects ten elements from 1 to 2 at random. The probability of getting element 1 is 25% and the probability of getting element 2 is 75%.

1.k

```
round(3.14159,digits = 2)

## [1] 3.14
```

Rounds the number to two digits after the decimal point.

1.l

```
range(100:400)

## [1] 100 400
```

Prints a vector with the minimum and maximum elements in the range 100 to 400.

1.m

```
matrix(c(1,2.3,2,3,4,8,12,43,-4,-1,9,14), nr=3,nc=4)

##      [,1] [,2] [,3] [,4]
## [1,]  1.0    3   12   -1
## [2,]  2.3    4   43    9
## [3,]  2.0    8   -4   14
```

Creates a matrix with 3 rows and 4 columns and fills it by column by default.

1.n

```
matrix(c(1,2.3,2,3,4,8,12,43,-4,-1,9,14),nr=3,nc=4,byrow = T)

##      [,1] [,2] [,3] [,4]
## [1,]    1  2.3    2    3
## [2,]    4  8.0   12   43
## [3,]   -4 -1.0    9   14
```

Creates a matrix with 3 rows and 4 columns and fills it by row.

1.o

```
x <-matrix(c(4,3,4,6,7,6),3,2)
rownames(x) <-c("row1","row2","row3")
colnames(x) <-c("col1","col2")
x

##      col1 col2
## row1    4    6
## row2    3    7
## row3    4    6
```

Creates a matrix of elements 4,5,4,6,7,6 with 3 rows and 2 columns and fills it by columns. Then adds a label to the rows and columns.

1.p

```
x <- rbind(c(1:4),c(5,8))
x

##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    8    5    8
```

Creates a matrix x by joining elements from 1 to 4 and elements 5 and 8 by rows. Since there are four elements in the first argument, it repeats 5 and 8 twice to match the elements in argument 1 (4 columns, 2 rows).

```
y <- cbind(c(1:4),c(5,8))
y
```

```
##      [,1] [,2]
## [1,]    1    5
## [2,]    2    8
## [3,]    3    5
## [4,]    4    8
```

Creates a matrix x by joining elements from 1 to 4 and elements 5 and 8 by column. Since there are four elements in the first argument, it repeats 5 and 8 twice to match the elements in argument 1 (2 columns, 4 rows).

1.q

```
y<-1:9
w<-2:10
z<-3:5
rbind(y,w,z)
```

```
##   [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## y    1    2    3    4    5    6    7    8    9
## w    2    3    4    5    6    7    8    9   10
## z    3    4    5    3    4    5    3    4    5
```

Creates a matrix of 3 rows and 9 columns by binding elements row wise. Row 1 has elements from 1 to 9, row 2 has elements from 2 to 10, row 3 has elements from 3 to 5 which are repeated to match the number of elements in row 1 or row 2.

1.r

```
m<-matrix(1:36,9,4)
m
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1   10   19   28
##  [2,]    2   11   20   29
##  [3,]    3   12   21   30
##  [4,]    4   13   22   31
##  [5,]    5   14   23   32
##  [6,]    6   15   24   33
##  [7,]    7   16   25   34
##  [8,]    8   17   26   35
##  [9,]    9   18   27   36
```

Creates a matrix of 9 rows and 4 columns with elements from 1 to 36 which are filled column wise.

```r
m[2,3]

## [1] 20
```

Prints the element of 2nd row and 3rd column.

```r
m[,3]

## [1] 19 20 21 22 23 24 25 26 27
```

Prints all the elements of column 3 in row format.

```r
m[2,]

## [1]  2 11 20 29
```

Prints all the elements of row 2.

```r
cbind(m[,3])

##       [,1]
##  [1,]   19
##  [2,]   20
##  [3,]   21
##  [4,]   22
##  [5,]   23
##  [6,]   24
##  [7,]   25
##  [8,]   26
##  [9,]   27
```

Prints elements of 3rd column of matrix m in column format.

```r
m[,-3]

##       [,1] [,2] [,3]
##  [1,]    1   10   28
##  [2,]    2   11   29
##  [3,]    3   12   30
##  [4,]    4   13   31
##  [5,]    5   14   32
##  [6,]    6   15   33
##  [7,]    7   16   34
##  [8,]    8   17   35
##  [9,]    9   18   36
```

Prints only the 1st, 2nd and 4th columns of matrix m. Doesn’t print 3rd column.

```r
m[-(3:8),2:4]

##      [,1] [,2] [,3]
## [1,]   10   19   28
```

```
## [2,]   11   20   29
## [3,]   18   27   36
```

Prints elements of 1st, 2nd, 9th rows ignoring the rows from 3 to 8 and prints elements of columns 2,3 and 4.

1.s

```
x<-cbind(x1=3,x2=c(4:1,2:5))
x

##      x1 x2
## [1,]  3  4
## [2,]  3  3
## [3,]  3  2
## [4,]  3  1
## [5,]  3  2
## [6,]  3  3
## [7,]  3  4
## [8,]  3  5
```

Creates a matrix with column labels x1 and x2. x1 column has element 3 for all of its corresponding rows. x2 has values of vector containing elements 4 to 1 and 2 to 5. The number of rows in x1 depends on x2 (number of rows of x2= 8)

```
dimnames(x)[[1]] <-letters[1:8]
x

##   x1 x2
## a  3  4
## b  3  3
## c  3  2
## d  3  1
## e  3  2
## f  3  3
## g  3  4
## h  3  5
```

Prints the matrix x with row labels having letters from a to h.

```
apply(x,2,mean,trim = .2)

## x1 x2
##  3  3
```

Applies the mean function on the columns of the vector x. Mean of first column is 3, since all rows of column 1 has the value 1. Mean of second column is 3, since there are 8 rows and total of x2 is 24. 20% trimmed mean is applied.

```
col.sums <-apply(x,2,sum)
col.sums
```

```
## x1 x2
## 24 24
```

Applies the sum function on the columns of the vector x. Total of first column is 3x8 = 24. Similar to the second column.

```
row.sums <-apply(x,1,sum)
row.sums

## a b c d e f g h
## 7 6 5 4 5 6 7 8
```

Applies the sum function on the rows of the vector x, first row has values 3 and 4, hence the sum is 7.

```
apply(x,2,sort)

##      x1 x2
## [1,]  3  1
## [2,]  3  2
## [3,]  3  2
## [4,]  3  3
## [5,]  3  3
## [6,]  3  4
## [7,]  3  4
## [8,]  3  5
```

Applies the sort function on the columns of vector x. First column has only values 3, so it is already sorted. Second column's values are sorted in ascending order.

Problem 2.

Write the corresponding R code for each of the following questions. Try to use the functions in dplyr package, if possible.

2.a Assign the value 15 to a variable x and create a vector y with the values [1, 2, 3, 10, 100]. Multiply those vectors component-wise and save the result in an object z. Calculate the sum of all elements in z.

```
x <- 15
x

## [1] 15

y <- c(1,2,3,10,100)
y

## [1]   1   2   3  10 100

z <- x * y
z
```

```
## [1]   15   30   45  150 1500

sum(z)

## [1] 1740
```

2.b Generate a sequence from 0 to 10 and a sequence from 5 to −5.

```
a <- 0:10
a

##  [1]  0  1  2  3  4  5  6  7  8  9 10

b <- 5:-5
b

##  [1]  5  4  3  2  1  0 -1 -2 -3 -4 -5
```

2.c Generate a sequence from −3 to 3 by 0.1 steps.

```
c <- seq(-3,3, by = 0.1)
c

##  [1] -3.0 -2.9 -2.8 -2.7 -2.6 -2.5 -2.4 -2.3 -2.2 -2.1 -2.0 -1.9 -1.8 -1.7
-1.6
## [16] -1.5 -1.4 -1.3 -1.2 -1.1 -1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2
-0.1
## [31]  0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0  1.1  1.2  1.3
1.4
## [46]  1.5  1.6  1.7  1.8  1.9  2.0  2.1  2.2  2.3  2.4  2.5  2.6  2.7  2.8
2.9
## [61]  3.0
```

2.d Define two vectors with the following data: t includes the strings “mon”, “tue”, “wed”, “thu”, “fri”, “sat”; and m includes [90, 80, 50, 20, 5, 20]. Concatenate both vectors column-wise into a matrix with 6 rows and 2 columns and save this a a new object named study.

```
t <- c("mon", "tue", "wed", "thu", "fri", "sat")
t

## [1] "mon" "tue" "wed" "thu" "fri" "sat"

m <- c(90,80,50,20,5,20)
m

## [1] 90 80 50 20  5 20

study <- cbind(t,m)
study

##      t     m   
## [1,] "mon" "90"
## [2,] "tue" "80"
```

```
## [3,] "wed" "50"
## [4,] "thu" "20"
## [5,] "fri" "5"
## [6,] "sat" "20"
```

2.e Create the following data frame:

age	sex	height	weight
21	m	181	69
35	f	173	58
829	m	171	75
2	e	166	60

Calculate the minimum and maximum value in the column age. Obviously, there have been some issues collecting the data. Generate a variable selection that contains the result to the logical query of age under 20 and above 80. Use this variable to set the age observations to NA if age is under 20 or above 80. Calculate the Body Mass Index (BMI)

```
BMI = Weight in kg/Length in m
```

of all people from the previous data frame. Store the results in a variable BMI and append it to your data frame. Round the resulting values.

```
age <- c(21,35,829,2)
sex <- c("m","f","m","e")
height <- c(181,173,171,166)
weight <- c(69,58,75,60)

df <- data.frame(age,sex,height,weight)
df

##   age sex height weight
## 1  21   m    181     69
## 2  35   f    173     58
## 3 829   m    171     75
## 4   2   e    166     60

print(min(df$age))

## [1] 2

print(max(df$age))

## [1] 829

selection <- df$age<20 | df$age>80
df$age[selection==TRUE] <- NA
df
```

```
##   age sex height weight
## 1  21   m    181     69
## 2  35   f    173     58
## 3  NA   m    171     75
## 4  NA   e    166     60
```

```
BMI <- round(df$weight/(df$height/100))
BMI
```

```
## [1] 38 34 44 36
```

Presuming height is in cm, we are converting to meter by dividing height by 100.

```
df1 <- cbind(df,BMI)
df1
```

```
##   age sex height weight BMI
## 1  21   m    181     69  38
## 2  35   f    173     58  34
## 3  NA   m    171     75  44
## 4  NA   e    166     60  36
```

Problem 3.

Set x to the following vector:

```
x <-c(9, 8, 12, 6, 1, 10, 10, 10, 8, 516, 8, 6, 4, 19, 100)
```

Provide the corresponding R function for each of the following task.

3.a Compute the mean of x.

```
mean(x)
```

```
## [1] 48.46667
```

3.b Compute the standard deviation of x.

```
sd(x)
```

```
## [1] 131.5261
```

3.c Compute the range of x.

```
range(x)
```

```
## [1]   1 516
```

3.d Provide the five number summary of x.

```
fivenum(x)
```

```
## [1]   1   7   9  11 516
```

3.e Is there any NA in x?

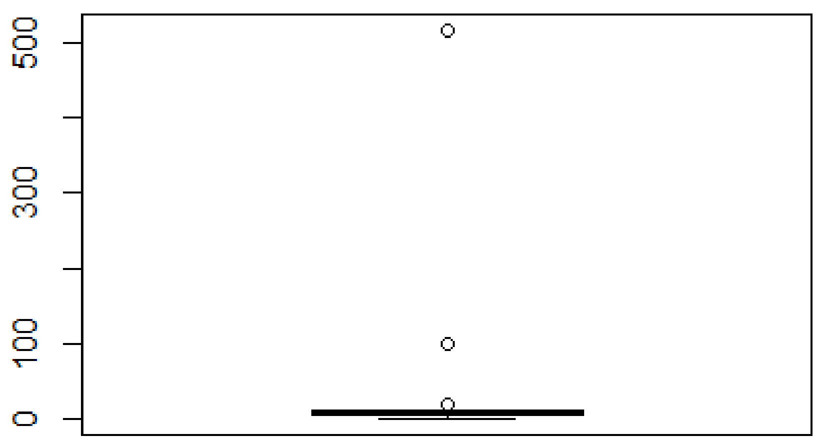
```
is.na(x)

##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FAL
SE
## [13] FALSE FALSE FALSE
```

There is no NA in x.

3.f Are there any outliers in x? If yes, remove them.

boxplot the vector x to check for outliers.

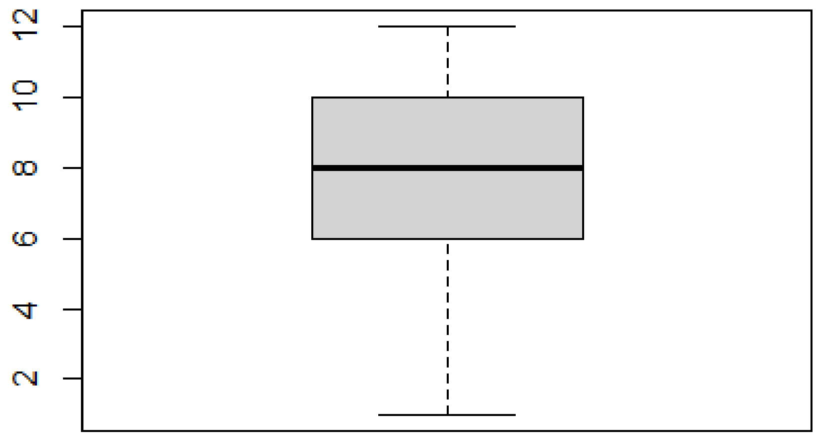
```
boxplot(x)
```



Yes, There are outliers present.

Removing outliers

```
outliers <- boxplot(x, plot=FALSE)$out
outliers

## [1] 516  19 100

x<- x[-which(x %in% outliers)]
boxplot(x)
```



Problem 4.

Consider the arbuthnot.csv dataset. This dataset refers to Dr. John Arbuthnot who was interested in the ratio of newborn boys to newborn girls. He gathered the baptism records for children born in London for every year from 1629 to 1710. Please include the corresponding R code you use to answer each of the questions below.

Reading arbuthnot.csv file

```
arbuthnot <- read.csv("arbuthnot.csv")
```

4.a What is the dimension of this dataset?

```
dim(arbuthnot)

## [1] 82  4
```

4.b What are the names of the variables in this dataset?

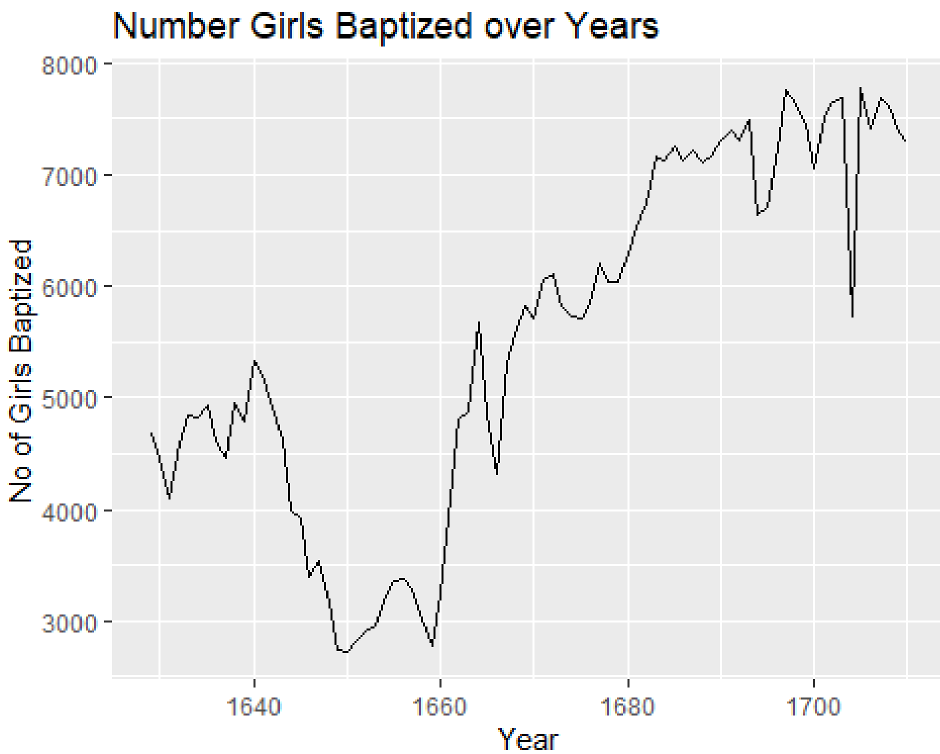
```
names(arbuthnot)

## [1] "X"     "year"  "boys"  "girls"
```

4.c What command would you use to extract just the counts of girls baptized?

```
length(arbuthnot$girls)

## [1] 82
```

4.d Is there an apparent trend in the number of girls baptized over the years? How would you describe it?

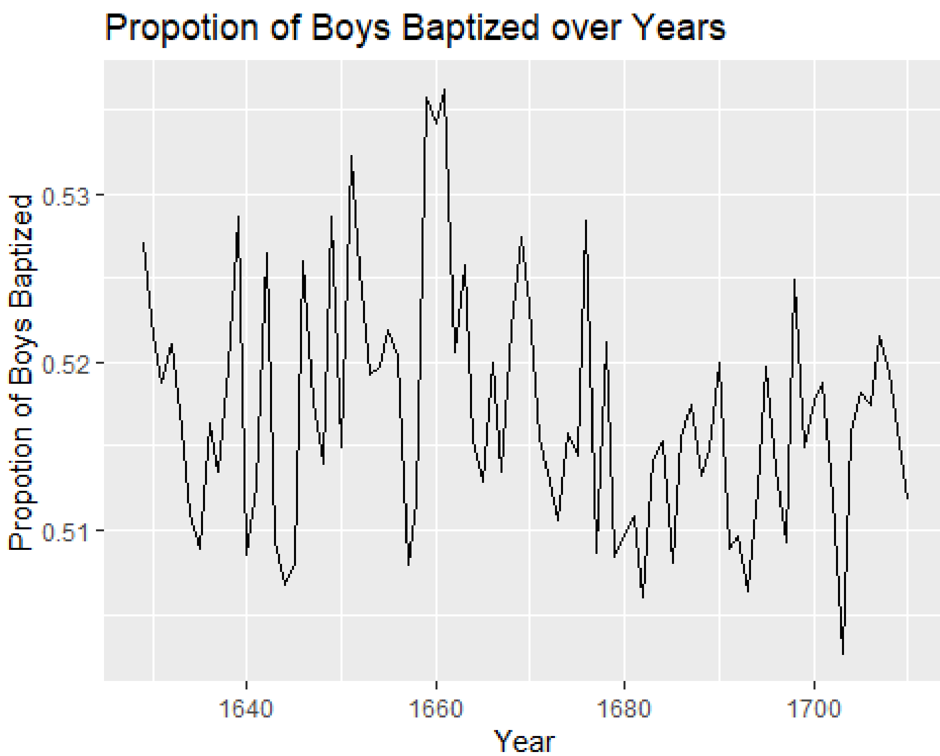
```
ggplot(data = arbuthnot, mapping = aes(x = year, y = girls)) + geom_line() +
  ggtitle("Number Girls Baptized over Years") + xlab("Year") +
  ylab("No of Girls Baptized")
```



There was an initial positive trend in the number of girls baptized over years till 1640, then there was a steep negative trend till 1660. Finally, the positive trend continued again. There seemed to be a steep down trend in 1703, which was quickly followed by a correction.

4.e Now, make a plot of the proportion of boys over time. What do you see?

```
p <- arbuthnot$boys / (arbuthnot$boys + arbuthnot$girls)

ggplot(data = arbuthnot, mapping = aes(x = year, y = p)) + geom_line() +
  ggtitle("Propotion of Boys Baptized over Years") + xlab("Year") +
  ylab("Propotion of Boys Baptized")
```



The proportion of boys Baptized overtime is generally between 50% and 53.5%.

4.f In what year did we see the most total number of births in the London?

```
arbuthnot %>% select(year) %>% filter(arbuthnot$boys == max(arbuthnot$boys))

##   year
## 1 1698
```

The most total number of births in London was in the year 1698.

Problem 5.

In this question, we use the built-in R dataset called attitude which contains information from a survey of the clerical employees of a large financial organization. To access this date set use "data("attitude")". Learn more about each variable by reading the variable description in ?attitude.

5.a Summarize the main statistics of all the variables in the data set.

```
att<-attitude

summary(att)

##      rating        complaints     privileges      learning         raises
##  Min.   :40.00   Min.   :37.0   Min.   :30.00   Min.   :34.00   Min.   :43
.00
##  1st Qu.:58.75   1st Qu.:58.5   1st Qu.:45.00   1st Qu.:47.00   1st Qu.:58
```

```
.25
##  Median :65.50   Median :65.0   Median :51.50   Median :56.50   Median :63
.50
##  Mean   :64.63   Mean   :66.6   Mean   :53.13   Mean   :56.37   Mean    64
.63
##  3rd Qu.:71.75   3rd Qu.:77.0   3rd Qu.:62.50   3rd Qu.:66.75   3rd Qu.:71
.00
##  Max.   :85.00   Max.   :90.0   Max.   :83.00   Max.   :75.00   Max.    88
.00
##     critical        advance
##  Min.   :49.00   Min.   :25.00
##  1st Qu.:69.25   1st Qu.:35.00
##  Median :77.50   Median :41.00
##  Mean   :74.77   Mean   :42.93
##  3rd Qu.:80.00   3rd Qu.:47.75
##  Max.   :92.00   Max.   :72.00
```

5.b How many observations are in the attitude dataset? What function in R did you use to display this information?

```
str(attitude)

## 'data.frame':    30 obs. of  7 variables:
##  $ rating    : num  43 63 71 61 81 43 58 71 72 67 ...
##  $ complaints: num  51 64 70 63 78 55 67 75 82 61 ...
##  $ privileges: num  30 51 68 45 56 49 42 50 72 45 ...
##  $ learning  : num  39 54 69 47 66 44 56 55 67 47 ...
##  $ raises    : num  61 63 76 54 71 54 66 70 71 62 ...
##  $ critical  : num  92 73 86 84 83 49 68 66 83 80 ...
##  $ advance   : num  45 47 48 35 47 34 35 41 31 41 ...
```

There are 30 observations in 'attitude' dataset. str() function in R returns information such as number of observations, number of variables.

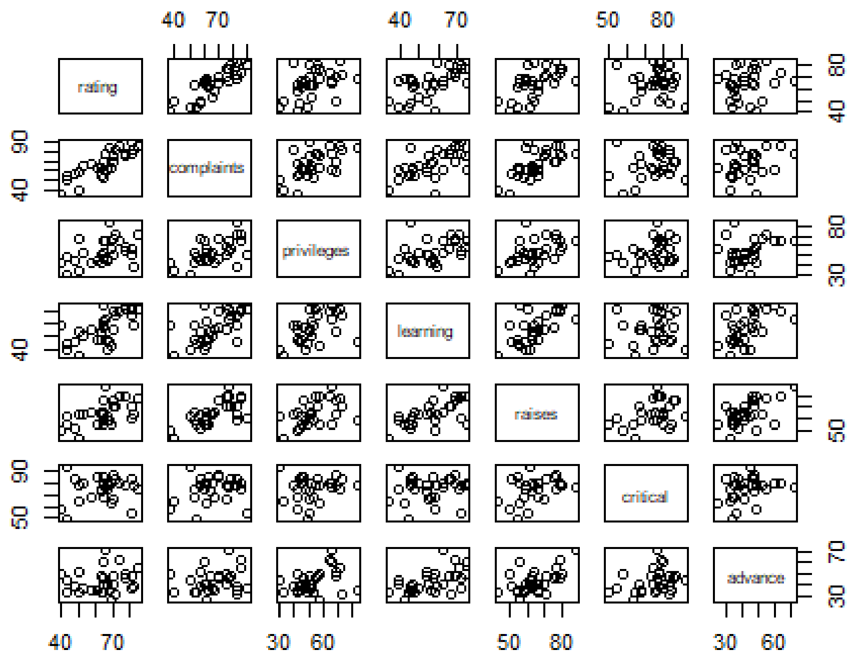
OR

```
nrow(attitude)

## [1] 30
```

nrow() function returns the number of rows which are the number of observations in the dataset.

5.c Produce a scatterplot matrix of the variables in the attitude dataset. What seems to be most correlated with the overall rating?

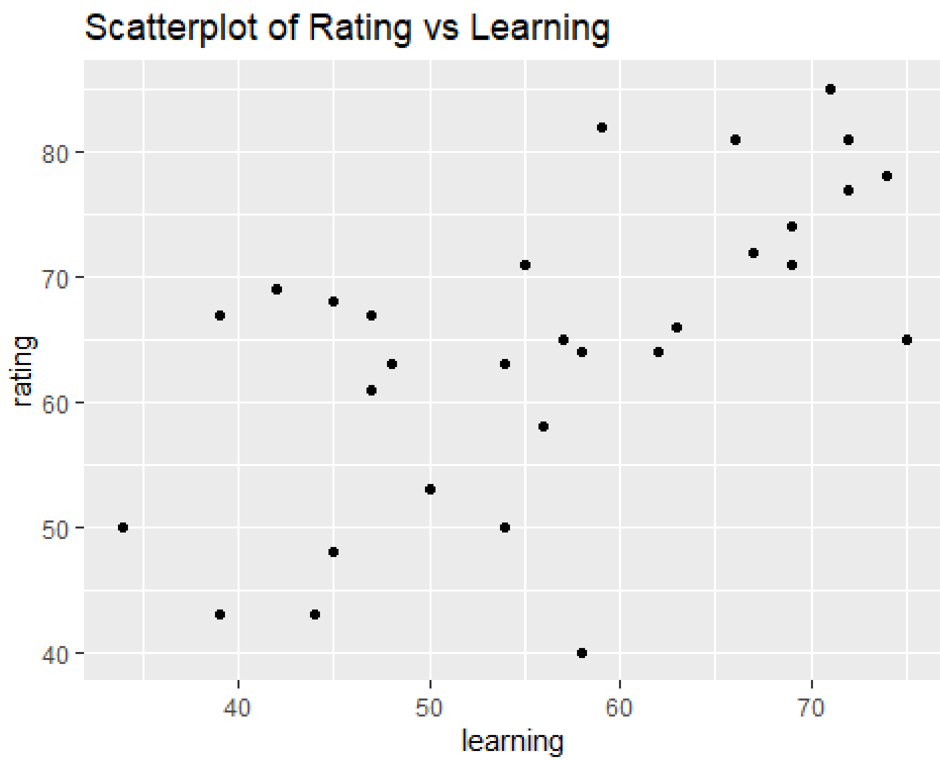
```
plot(attitude)
```



Based on the graph, Complaints seems to be the most correlated with the overall rating. We also used the function cor() to verify the correlation between variables.

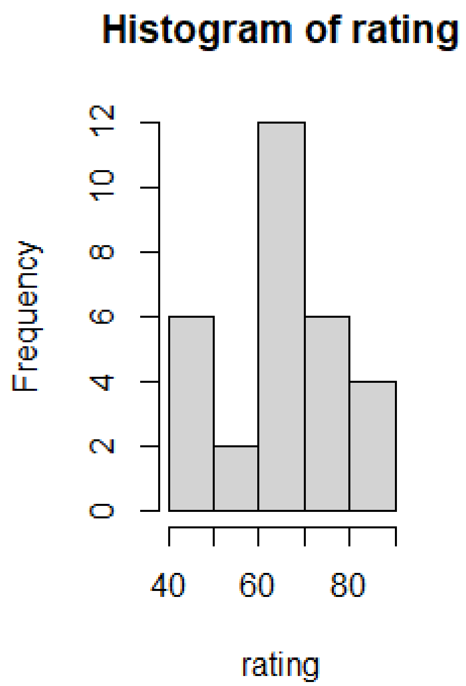
5.d Produce a scatterplot of rating (on the y-axis) vs. learning (on the x-axis). Add a title to the plot.

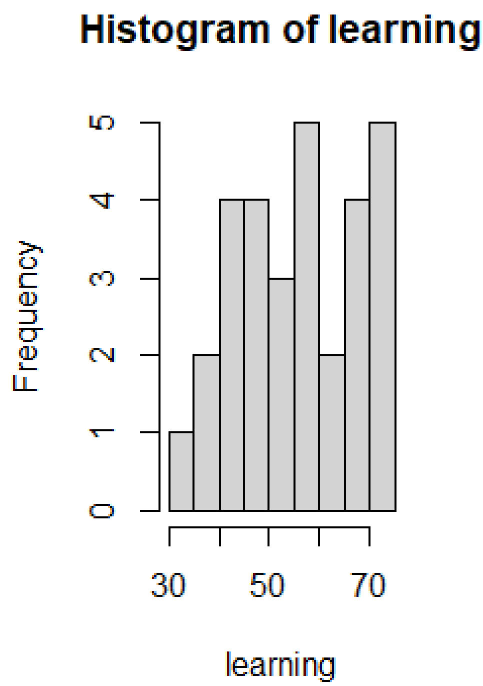
```
ggplot (att, mapping = aes(x=learning,y=rating)) +
  geom_point()+
  ggtitle("Scatterplot of Rating vs Learning")
```



5.e Produce 2 side-by-side histograms, one for rating and one for learning. You will need to use par(mfrow=...) to get the two plots together.

```
rating <- att$rating
learning <- att$learning
par(mfrow = c(1, 2))
hist(rating)
hist(learning)
```





Problem 6.

Write the corresponding R code for each of the following questions. Try to use the functions in dplyr package, if possible.

6.a In one or two lines describe what this data set is about. What variables are included in this dataset (look at the help: ?mtcars)?

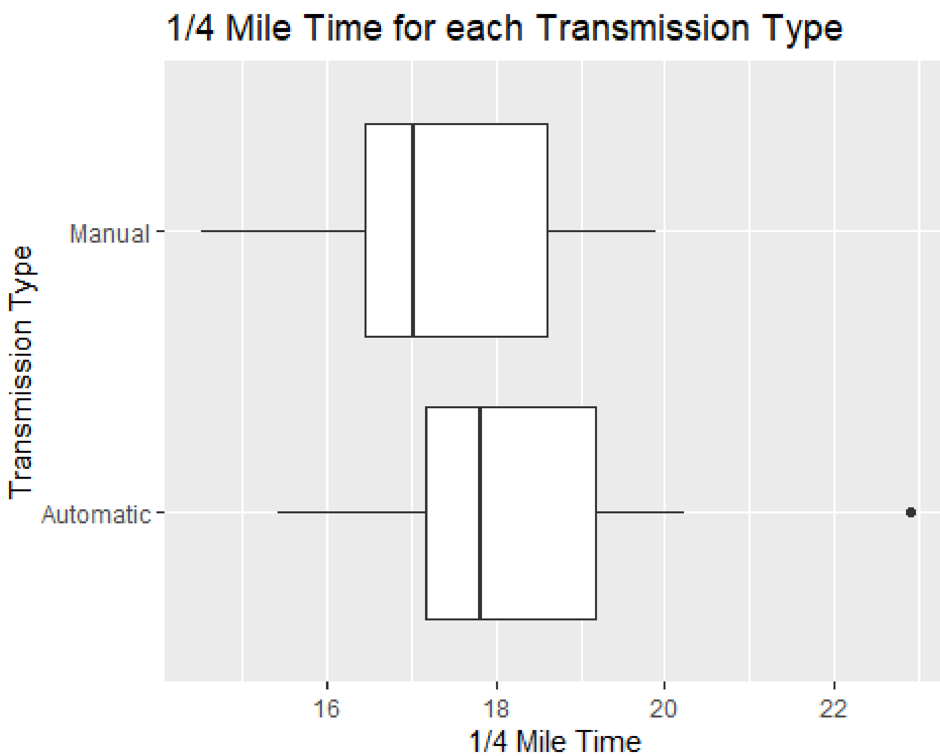
mtcars dataset is obtained from the 1974 Motor Trend US magazine. The dataset consists of data such as fuel consumption and 10 aspects of car design for 32 automobiles.

The are 11 numeric variables in this dataset:

mpg	Miles/(US) gallon
cyl	Number of cylinders
disp	Displacement (cu.in.)
hp	Gross horsepower
drat	Rear axle ratio
wt	Weight (1000 lbs)
qsec	1/4 mile time
vs	Engine (0 = V-shaped, 1 = straight)
am	Transmission (0 = automatic, 1 = manual)
gear	Number of forward gears
carb	Number of carburetors

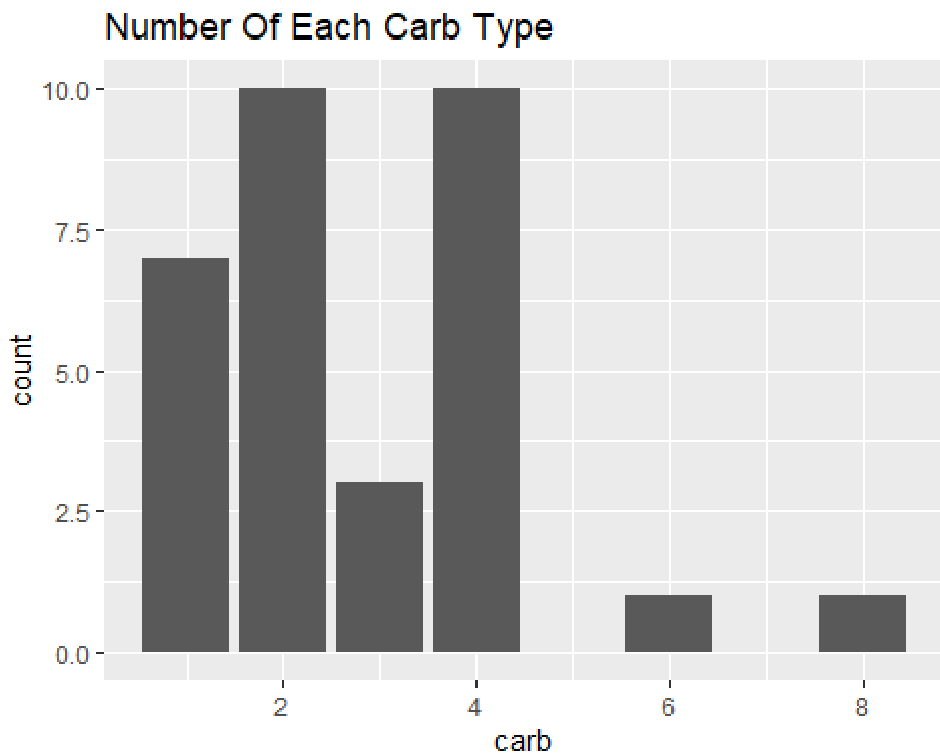
6.b Create a box plot using ggplot showing the range of values of 1/4 mile time (qsec) for each tansmission type (am, 0 = automatic, 1 = manual) from the mtcars data set. Use “Transmission Type” and “1/4 Mile Time” for your y- and x-axes respectively. Also, add the title to your graph.

```
cars <- mtcars

cars$am = factor(cars$am, levels=c(0,1), labels=c("Automatic","Manual"))
cars %>%
  ggplot(mapping = aes(x = qsec, y = am)) +
  geom_boxplot() +
  ggtitle("1/4 Mile Time for each Transmission Type") +
  xlab("1/4 Mile Time") +
  ylab("Transmission Type")
```



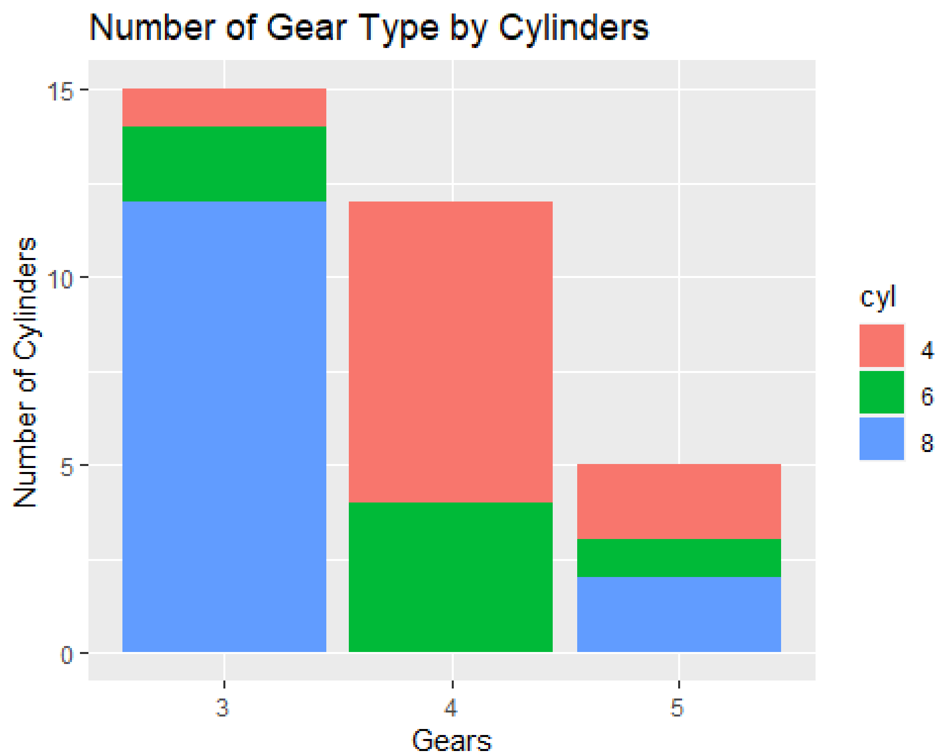
6.c Create a bar graph using ggplot, that shows the number of each carb type in mtcars.

```
cars %>%
  ggplot(aes(carb)) +
  geom_bar() +
  ggtitle("Number Of Each Carb Type")
```



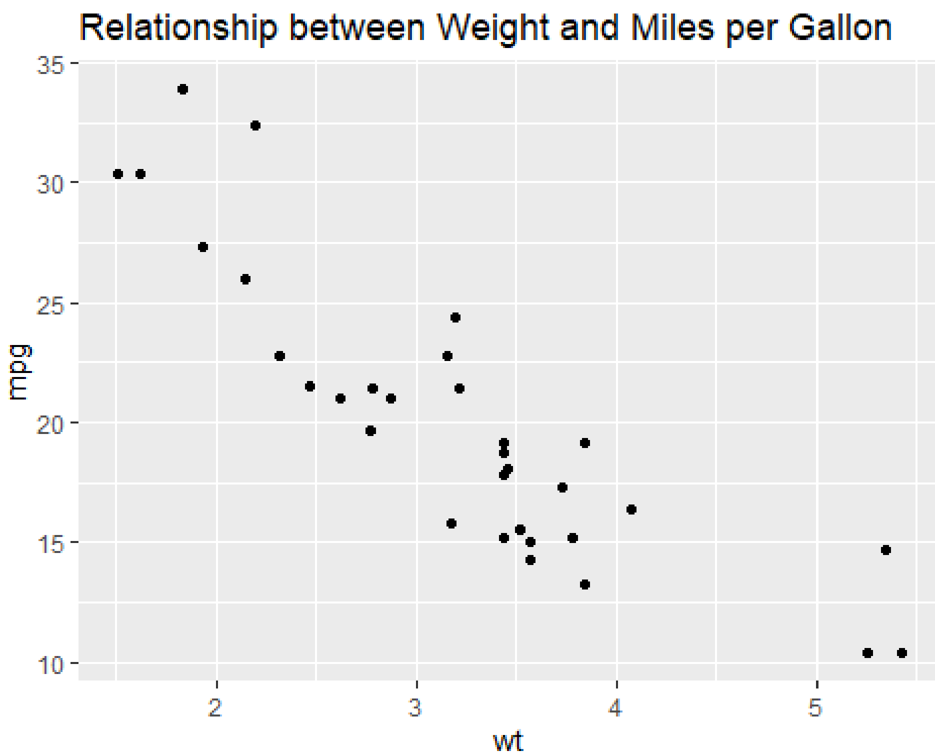
6.d Next show a stacked bar graph using ggplot of the number of each gear type and how they are further divided out by cyl. Add labels and a title to your plot.

```
cars$gear = factor(cars$gear)
cars$cyl = factor(cars$cyl)

cars %>%
  ggplot(aes(x = gear, fill = cyl)) +
  geom_bar() +
  ggtitle ("Number of Gear Type by Cylinders")+
  xlab("Gears") +
  ylab("Number of Cylinders")
```



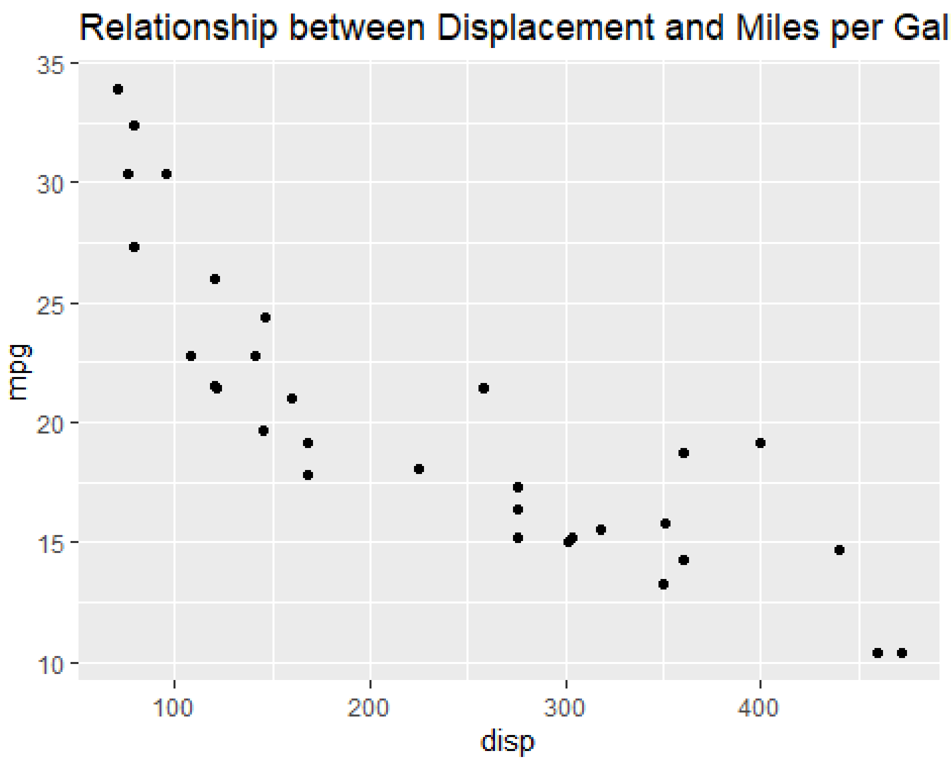
6.e Draw a scatter plot using ggplot showing the relationship between wt and mpg.

```
cars %>%
  ggplot(aes(x=wt, y=mpg)) +
  geom_point() +
  ggtitle ("Relationship between Weight and Miles per Gallon")
```



6.f Draw a scatter plot to investigate the relationship between “disp” and “mpg”. What do you observe. Explain.

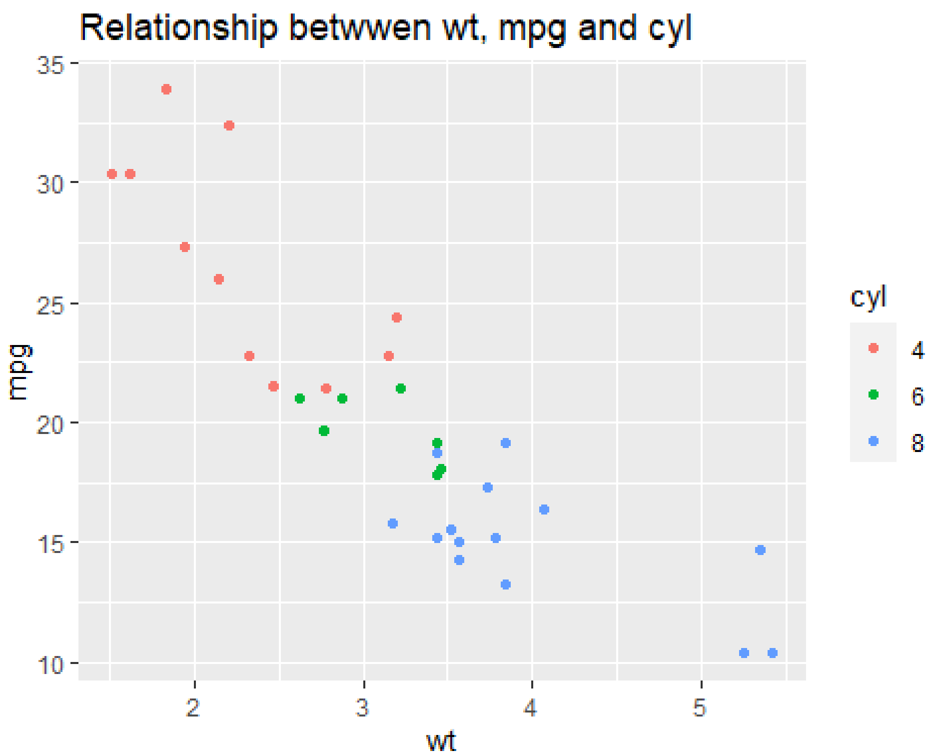
```
cars %>%
  ggplot(aes(x=disp, y=mpg)) +
  geom_point() +
  ggtitle ("Relationship between Displacement and Miles per Gallon")
```



Observation: With an increase in the displacement, the fuel efficiency measured in miles per gallon is decreased.

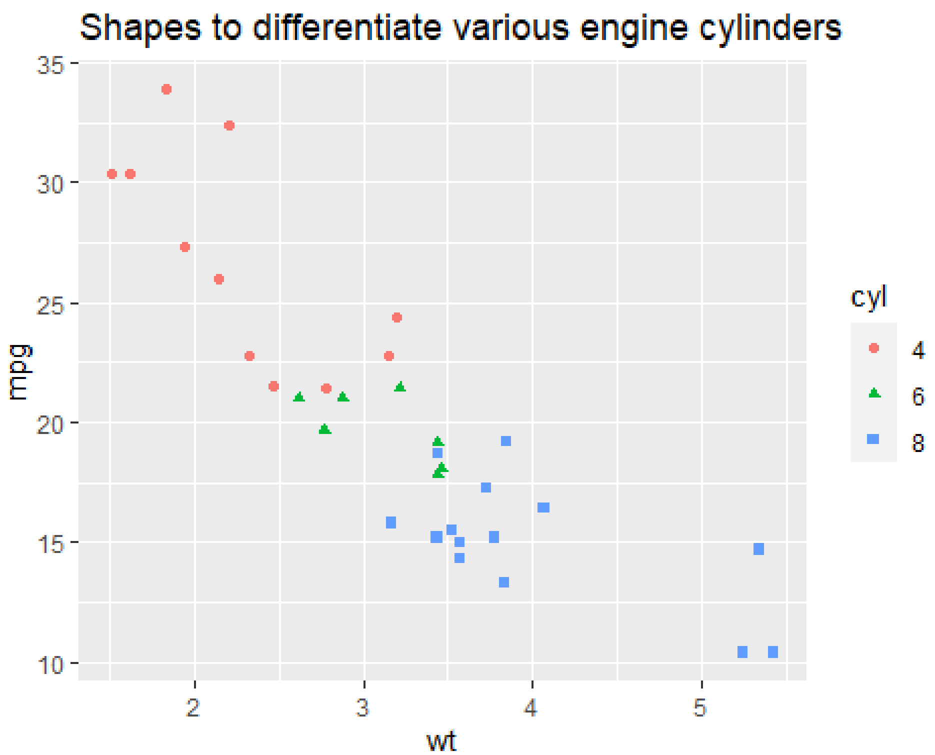
6.g Create a scatter plot that shows the relationship between various car weights (wt), miles per gallon (mpg) and engine cylinders (cyl). Use colored points to show the different cylinders in the plot. Note: you will need to convert cyl to a factor. You will need the function factor() to do this.

```
cars$cyl <- factor(cars$cyl)
cars %>%
  ggplot(aes(x=wt, y=mpg))+
  geom_point(aes(color=cyl)) +
  ggtitle ("Relationship betwwen wt, mpg and cyl")
```



6.h Using the solution from part (g), create a new plot using shapes to differentiate the various engine cylinders.

```
cars %>%
  ggplot(aes(x=wt, y=mpg))+
  geom_point(aes(color=cyl, shape=cyl)) +
  ggtitle ("Shapes to differentiate various engine cylinders")
```



Problem 7.

Download the gapminder.csv data and read it into R. Assign the data to an object called gm. Use this dataset to answer the following questions. Try to use the functions in dplyr package if possible.

Reading gapminder.csv file

```
gm <- read.csv("gapminder.csv")
```

7.a How many unique countries are represented per continent?

```
gm %>% group_by(continent) %>% summarize(unique_countries = n_distinct(countr
y))
```

```
## # A tibble: 5 x 2
##   continent unique_countries
##   <chr>                <int>
## 1 Africa                  52
## 2 Americas                25
## 3 Asia                    33
## 4 Europe                  30
## 5 Oceania                  2
```

7.b Which European nation had the lowest GDP per capita in 1997?

```
gm %>% filter(continent == "Europe" , year == 1997) %>% arrange(gdpPercap) %>%
%
  head(1)

##   country continent year lifeExp     pop gdpPercap
## 1 Albania    Europe 1997   72.95 3428038  3193.055
```

Albania had the lowest GDP per capita in 1997.

7.c According to the data available, what was the average life expectancy across each continent in the 1980s?

```
gm %>% group_by(continent) %>% filter(year >= 1980, year <= 1989) %>%
  summarize(mean_lifeExp = mean(lifeExp))

## # A tibble: 5 x 2
##   continent mean_lifeExp
##   <chr>            <dbl>
## 1 Africa            52.5
## 2 Americas          67.2
## 3 Asia              63.7
## 4 Europe            73.2
## 5 Oceania           74.8
```

7.d What 5 countries have the highest total GDP over all years combined?

GDP is a measure that results from GDP per capita multiplied by the size of the nation's overall population

```
gm %>% mutate(gdp = gdpPercap*pop) %>% group_by(country) %>%
  summarise(Total.GDP = sum(gdp)) %>% arrange(desc(Total.GDP)) %>% head(5)

## # A tibble: 5 x 2
##   country        Total.GDP
##   <chr>              <dbl>
## 1 United States    7.68e13
## 2 Japan            2.54e13
## 3 China            2.04e13
## 4 Germany          1.95e13
## 5 United Kingdom   1.33e13
```

United States, Japan, China, Germany and United Kingdom are the 5 countries having highest total GDP over all years combined.

7.e What countries and years had life expectancies of at least 80 years? N.b. only output the columns of interest: country, life expectancy and year (in that order).

```
gm %>% select(country, lifeExp, year) %>% filter(lifeExp >= 80)
```

```
##             country lifeExp year
## 1        Australia  80.370 2002
## 2        Australia  81.235 2007
## 3           Canada  80.653 2007
## 4           France  80.657 2007
## 5  Hong Kong, China  80.000 1997
## 6  Hong Kong, China  81.495 2002
## 7  Hong Kong, China  82.208 2007
## 8          Iceland  80.500 2002
## 9          Iceland  81.757 2007
## 10          Israel  80.745 2007
## 11           Italy  80.240 2002
## 12           Italy  80.546 2007
## 13           Japan  80.690 1997
## 14           Japan  82.000 2002
## 15           Japan  82.603 2007
## 16     New Zealand  80.204 2007
## 17          Norway  80.196 2007
## 18           Spain  80.941 2007
## 19          Sweden  80.040 2002
## 20          Sweden  80.884 2007
## 21     Switzerland  80.620 2002
## 22     Switzerland  81.701 2007
```

Problem 8.

To answer this question we use R built in data set “hflights” from hflights package. Write the corresponding R code to to answer the following questions. Try se the functions in dplyr package if possible.

8.a Look at the first 20 instances in your data set.

```
flights <- hflights

head(hflights,20)

##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightN
um
## 5424 2011     1          1         6    1400    1500            AA       4
28
## 5425 2011     1          2         7    1401    1501            AA       4
28
## 5426 2011     1          3         1    1352    1502            AA       4
28
## 5427 2011     1          4         2    1403    1513            AA       4
28
## 5428 2011     1          5         3    1405    1507            AA       4
28
## 5429 2011     1          6         4    1359    1503            AA       4
28
```

```
## 5430 2011     1          7         5    1359    1509            AA       4
28
## 5431 2011     1          8         6    1355    1454            AA       4
28
## 5432 2011     1          9         7    1443    1554            AA       4
28
## 5433 2011     1         10         1    1443    1553            AA       4
28
## 5434 2011     1         11         2    1429    1539            AA       4
28
## 5435 2011     1         12         3    1419    1515            AA       4
28
## 5436 2011     1         13         4    1358    1501            AA       4
28
## 5437 2011     1         14         5    1357    1504            AA       4
28
## 5438 2011     1         15         6    1359    1459            AA       4
28
## 5439 2011     1         16         7    1359    1509            AA       4
28
## 5440 2011     1         17         1    1530    1634            AA       4
28
## 5441 2011     1         18         2    1408    1508            AA       4
28
## 5442 2011     1         19         3    1356    1503            AA       4
28
## 5443 2011     1         20         4    1507    1622            AA       4
28
##      TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin Dest Dista
nce
## 5424  N576AA                60      40      -10        0    IAH  DFW      
224
## 5425  N557AA                60      45       -9        1    IAH  DFW      
224
## 5426  N541AA                70      48       -8       -8    IAH  DFW      
224
## 5427  N403AA                70      39        3        3    IAH  DFW      
224
## 5428  N492AA                62      44       -3        5    IAH  DFW      
224
## 5429  N262AA                64      45       -7       -1    IAH  DFW      
224
## 5430  N493AA                70      43       -1       -1    IAH  DFW      
224
## 5431  N477AA                59      40      -16       -5    IAH  DFW      
224
## 5432  N476AA                71      41       44       43    IAH  DFW      
224
## 5433  N504AA                70      45       43       43    IAH  DFW      
224
```

```
## 5434  N565AA                 70      42        29        29    IAH  DFW
224
## 5435  N577AA                 56      41         5        19    IAH  DFW
224
## 5436  N476AA                 63      44        -9        -2    IAH  DFW
224
## 5437  N552AA                 67      47        -6        -3    IAH  DFW
224
## 5438  N462AA                 60      44       -11        -1    IAH  DFW
224
## 5439  N555AA                 70      41        -1        -1    IAH  DFW
224
## 5440  N518AA                 64      48        84        90    IAH  DFW
224
## 5441  N507AA                 60      42        -2         8    IAH  DFW
224
## 5442  N523AA                 67      46        -7        -4    IAH  DFW
224
## 5443  N425AA                 75      42        72        67    IAH  DFW
224
##      TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 5424      7      13         0                        0
## 5425      6       9         0                        0
## 5426      5      17         0                        0
## 5427      9      22         0                        0
## 5428      9       9         0                        0
## 5429      6      13         0                        0
## 5430     12      15         0                        0
## 5431      7      12         0                        0
## 5432      8      22         0                        0
## 5433      6      19         0                        0
## 5434      8      20         0                        0
## 5435      4      11         0                        0
## 5436      6      13         0                        0
## 5437      5      15         0                        0
## 5438      6      10         0                        0
## 5439     12      17         0                        0
## 5440      8       8         0                        0
## 5441      7      11         0                        0
## 5442     10      11         0                        0
## 5443      9      24         0                        0
```

8.b View all flights on January 1st.

```
flights %>% filter(Month == 1, DayofMonth == 1) %>% head(5)
```

```
##   Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 1 2011     1          1         6    1400    1500            AA       428
## 2 2011     1          1         6     728     840            AA       460
## 3 2011     1          1         6    1631    1736            AA      1121
```

```
## 4 2011     1          1         6    1756    2112            AA      1294
## 5 2011     1          1         6    1012    1347            AA      1700
##   TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin Dest Distance
## 1  N576AA                60      40      -10        0    IAH  DFW      224
## 2  N520AA                72      41        5        8    IAH  DFW      224
## 3  N4WVAA                65      37       -9        1    IAH  DFW      224
## 4  N3DGAA               136     113       -3        1    IAH  MIA      964
## 5  N3DAAA               155     117        7       -8    IAH  MIA      964
##   TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 1      7      13         0                         0
## 2      6      25         0                         0
## 3     16      12         0                         0
## 4      9      14         0                         0
## 5     12      26         0                         0
```

We are printing only the top 5 rows of the result set.

8.c Only view the part of the dataset that is related to American or United Airlines carriers.

```
flights %>% filter(UniqueCarrier=="AA" | UniqueCarrier=="UA") %>% head(5)
```

```
##   Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 1 2011     1          1         6    1400    1500            AA       428
## 2 2011     1          2         7    1401    1501            AA       428
## 3 2011     1          3         1    1352    1502            AA       428
## 4 2011     1          4         2    1403    1513            AA       428
## 5 2011     1          5         3    1405    1507            AA       428
##   TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin Dest Distance
## 1  N576AA                60      40      -10        0    IAH  DFW      224
## 2  N557AA                60      45       -9        1    IAH  DFW      224
## 3  N541AA                70      48       -8       -8    IAH  DFW      224
## 4  N403AA                70      39        3        3    IAH  DFW      224
## 5  N492AA                62      44       -3        5    IAH  DFW      224
##   TaxiIn TaxiOut Cancelled CancellationCode Diverted
## 1      7      13         0                         0
## 2      6       9         0                         0
## 3      5      17         0                         0
## 4      9      22         0                         0
## 5      9       9         0                         0
```

We are printing only the top 5 rows of the result set.

8.d Look at a subset of your dataset that contains the variables “Year, Month, DayofMonth” and any other variables that contains the words “Taxi” and “Delay”.

```
flights %>% select(Year, Month, DayofMonth, contains("Taxi"), contains("Delay
")) %>% head(5)
```

```
##      Year Month DayofMonth TaxiIn TaxiOut ArrDelay DepDelay
## 5424 2011     1          1      7      13      -10        0
## 5425 2011     1          2      6       9       -9        1
```

```
## 5426 2011     1          3      5      17       -8       -8
## 5427 2011     1          4      9      22        3        3
## 5428 2011     1          5      9       9       -3        5
```

We are printing only the top 5 rows of the result set.

8.e Print a subset of your dataset that includes the following variables “Departure Time”, “Arrivales Time” and “Flight Number”.

```
flights %>%
  rename("Departure Time" = DepTime, "Arrivales Time" = ArrTime, "Flight Numb
er" = FlightNum) %>%
  select("Departure Time", "Arrivales Time", "Flight Number") %>% head(5)
```

```
##      Departure Time Arrivales Time Flight Number
## 5424           1400           1500           428
## 5425           1401           1501           428
## 5426           1352           1502           428
## 5427           1403           1513           428
## 5428           1405           1507           428
```

We are printing only the top 5 rows of the result set.

8.f Print all the aircraft carriers whose departure time is delayed more than 60 minutes.

```
x <- flights %>% select(UniqueCarrier) %>% filter(flights$DepDelay>60)
unique(x)
```

```
##     UniqueCarrier
## 1              AA
## 10             AS
## 11             B6
## 16             CO
## 179            DL
## 185            OO
## 213            UA
## 215            US
## 221            WN
## 244            EV
## 255            F9
## 256            FL
## 258            MQ
## 419            XE
```

8.g Look at the carriers with their departure delays and sort them based on their departure delays.

```
arrange(flights, DepDelay) %>% select(UniqueCarrier, DepDelay) %>% head(5)
```

```
##   UniqueCarrier DepDelay
## 1            OO      -33
## 2            MQ      -23
```

```
## 3                 XE        -19
## 4                 XE        -19
## 5                 CO        -18
```

We are printing only the top 5 rows of the result set.

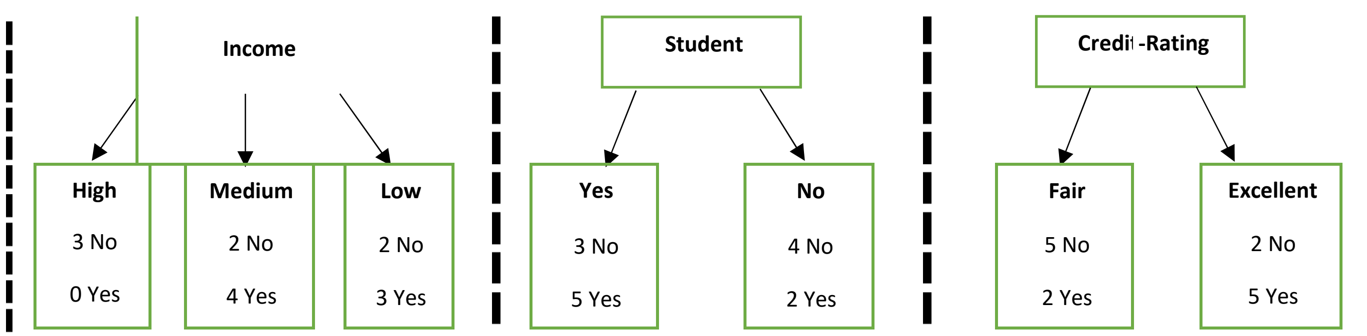
Problem 9. Consider the following data set:

Record Number	Income	Student	Credit Rating	Buys-computer
1.	High	No	Fair	No
2.	High	No	excellent	No
3.	Low	No	excellent	Yes
4.	Medium	No	Fair	no
5.	Low	Yes	Fair	no
6.	Low	Yes	excellent	yes
7.	Low	No	excellent	yes
8.	Medium	Yes	Fair	yes
9.	Low	Yes	Fair	No
10.	Medium	Yes	Fair	yes
11.	Medium	Yes	excellent	yes
12.	Medium	No	excellent	no
13.	High	Yes	Fair	no
14.	Medium	Yes	excellent	yes

a) Using the 1-rule method discussed in class, find the relevant sets of classification rules for the target buys-computer by testing each of the input attributes income, student, and credit-rating. Which of these three sets of rules has the lowest misclassification rate?

Solution: 1R method: Decisions are made on only one variable/attribute. Therefore, we would choose each variable and understand the sets for the target buys-computer.

Income has 3 values in the dataset. High, Medium & Low, similarly student has yes & no, and credit-rating has fair and excellent.



	Income	Student	Credit-Rating
Error	High: 0/3 Medium: 2/6 Low: 2/5	Yes: 3/8 No: 2/6	Fair: 2/7 Excellent: 2/7
Total Error	(0/3)*(3/14) + (2/6)*(6/14) + (2/5)*(5/14) = **2/7 = 0.285**	(3/8)*(8/14) + (2/6)*(6/14) = **5/14 = 0 357**	(2/7)*(7/14) + (2/7)*(7/14) **= 2/7 = 0.285**

Based on above calculations, the lowest misclassification rate (error) is for income and credit-rating.

b) 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 Gini index impurity measure?

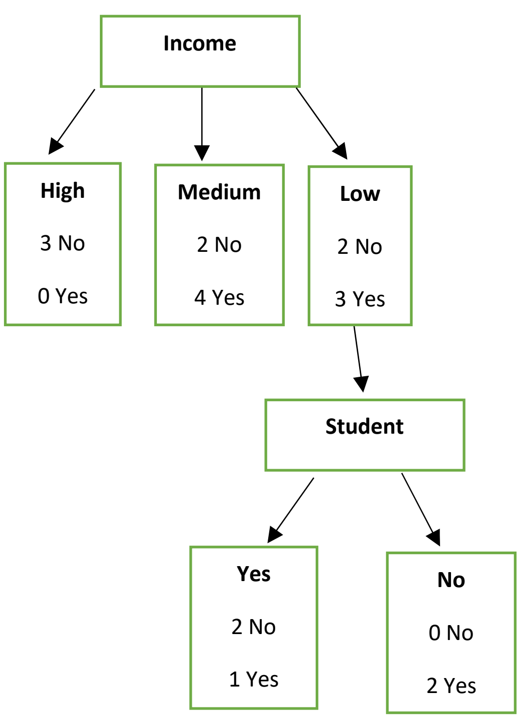
Solution:

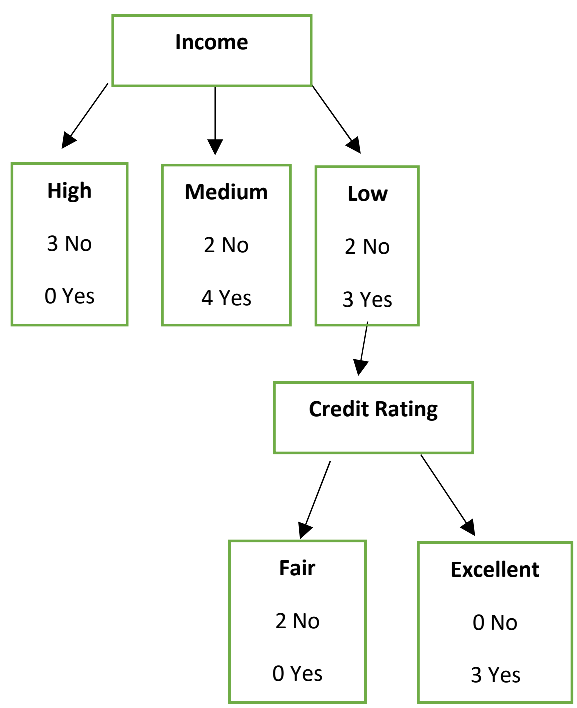
	Income	Student	Credit-Rating
Gini	High: $1-(0/3)^2-(3/3)^2 = 0$	No: $1-(2/6)^2-(4/6)^2 = 0.45$	Fair: $1-(2/7)^2-(5/7)^2 = 0.4$
	Low: $1-(3/5)^2-(2/5)^2 = 0.48$	Yes: $1-(5/8)^2-(3/8)^2 = 0.47$	Excellent: $1-(5/7)^2-(2/7)^2 = 0.4$
	Medium: $1-(4/6)^2-(2/6)^2 = 0.44$	Student: $(6/14)*(0.45)+(8/14)*(0.47)$	$(7/14)*(0.4)+(7/14)*(0.4)$ **= 0.4**
	Income: $(5/14)*(0.48)+(6/14)*(0.44)+0$ = **0.36**	**= 0.46**	

Based on the smallest Gini value (Income), the decision tree will be formed and further split. Income is the root node.

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

Solution : For income, high is a pure set. Hence no split is required. Low & Medium are impure sets therefore its needs to be further split. This will be judged by finding the Gini index of Student for Low and Medium. Also, Gini index of Credit-Rating for Low and Medium.





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

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

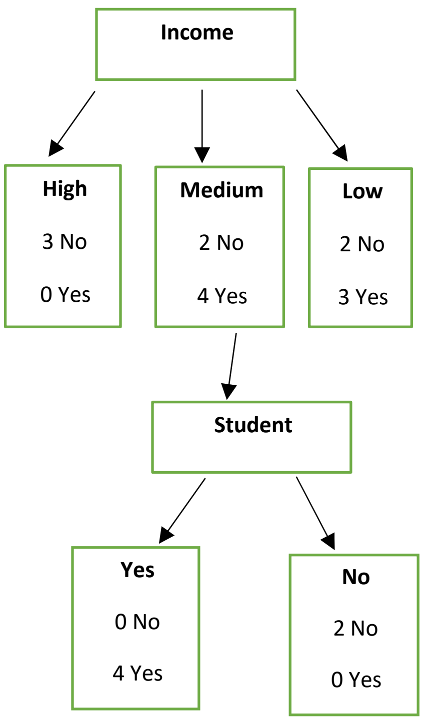
Gini (Student) = $(2/5) * 0 + (3/5) * 0.44 = 0.26$

Gini (fair) = $1 - (0/2)^2 - (2/2)^2 = 0$

Gini (excellent) = $1 - (3/3)^2 - (0/3)^2 = 0$

Gini (Credit) = $(2/5) * 0 + (3/5) * 0 = 0$

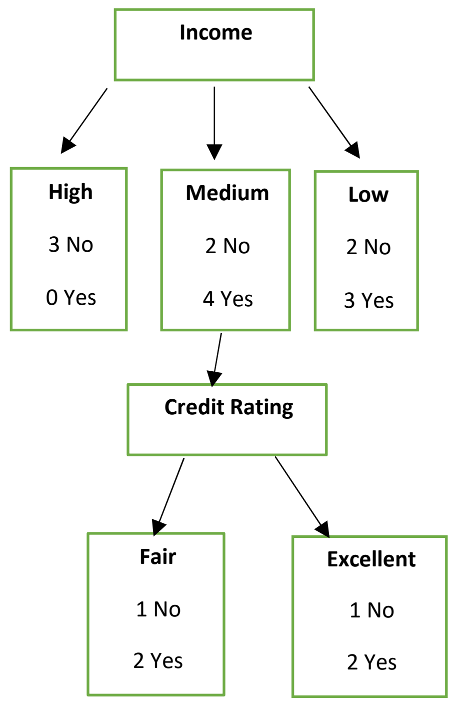
As per above calculations, the Gini value for Credit Rating is 0, which implies that it is a pure set. No further split is required. Hence, credit-rating is chosen as the next node.



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

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

Gini (Student) = $(2/6) * 0 + (4/6) * 0 = 0$

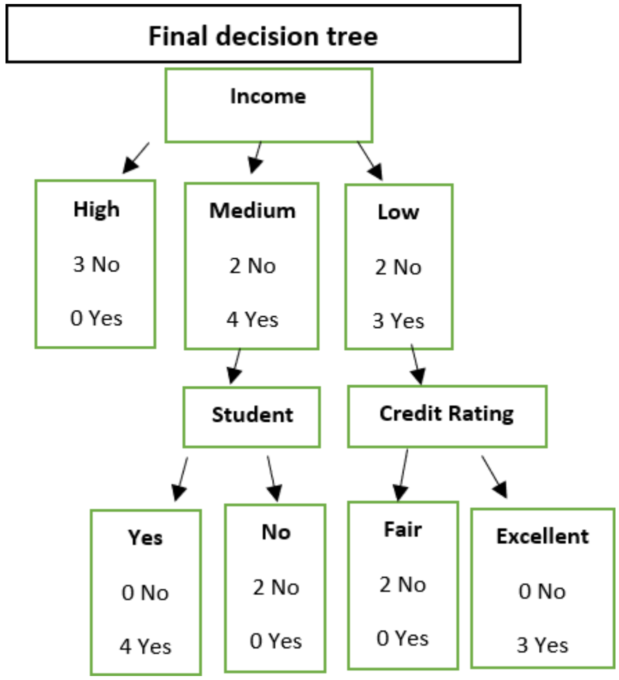


Gini (fair) = $1 - (2/3)^2 - (1/3)^2 = 0.44$

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

Gini (Credit) = $(3/6) * 0.44 + (3/6) * 0.44 = 0.44$

As per above calculations, the Gini value for Student is 0, which implies that it is a pure set. No further split is required. Hence, Student is chosen as the next node.



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

Solution: Based on the above decision tree the two decision rules are:

1: **If the income is high then a computer is not bought.**

Support: 3/14
Confidence: 3/3 = 100%

2: **If the income is medium and is a student then a computer is bought.**

Support: 4/14
Confidence: 4/4 = 100%

As the confidence is 100% for the above two decision rules, they are used to predict whether a student is going to buy computer or not.

e) What is the accuracy of your decision tree model on the training examples?

The accuracy of the decision tree is **100%** on the training examples as the decision tree rules hold good on all the instances of the training data.