Title: "IDS 572 HW1"
Group Members: "Navya Yadagiri","Sayali Bonawale","Aditya Madhavi"
UIN : 674788385, 656488690, 660225279
date: "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)
```

Values are stored in vector x

1. b)

```
max(x)
```

```
## [1] 43
```

This function returns maximum number stored in x.

1. c)

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

Values of vector x is stored in vector y and NA is appended at the last.

1. d)

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

```
## [1] 43
```

This function returns maximum number stored in y. Since, na.rm = TRUE, the function removes NA values.

1. e)

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

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

Values are stored in x2 vector and then min function returns the minimum value between the two vectors x and x2.

1. f)

```
sample(4:10)
```

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

This function generates random numbers between 4 and 10.

1. g)

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

```
## [1] 5 2 3
```

This function creates a group of size 3 by taking random samples from the vector. There is no repetition of numbers because the replacement parameter is FALSE.

1. h)

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

```
## [1] 3 5 5
```

Because the replacement parameter is set to TRUE, this function creates random samples from the vector and forms a group of size 3 with number repetition.

1. i)

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

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

This function will generate a group of size 10 which includes numbers 1 and 2 with repetition of these numbers. When the replace = FALSE, it gives an error since the sample size is larger than the dataset.

1. j)

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

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

This function will generate a group by size 10 with replacement which includes number 1 and 2 with the probability of 1 occurring is 10% and 2 is 30%.

1. k)

```r
round(3.14159, digits = 2)
```

```
## [1] 3.14
```

The number gets rounded up to two digits after the decimal with this function.

1. l)

```r
range(100:400)
```

```
## [1] 100 400
```

This function returns a vector with the minimum (inclusive 100) and maximum (inclusive 400) values within a data vector.

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
```

This will construct a three-rows(nr), four-columns(nc) matrix with the specified values. The matrix is filled column-wise.

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
```

This will construct a three-rows(nr), four-column matrix(nc) with the specified values and since, the byrow is equal to TRUE, the matrix will get filled row-wise.

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
```

The preceding code will generate a three-row, two-column matrix with the specified vector.

Row and column headers are assigned using rownames(x) and colnames(x).

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
```

The row bind combines the vectors provided row-wise. The first vector has values 1 to 4 and second vector has values 5 and 8. So it creates a matrix of 2 rows and 4 columns.

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

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

The column bind combines the vectors provided column-wise. The first vector has values 1 to 4 and second vector has values 5 and 8. So it creates a matrix of 4 rows and 2 columns.

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
```

The y, w, and z vectors will be created with the values specified. The row-bind will combine values row-wise with 9 columns since the y vector contains 9 elememts.

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
```

This will create a 9 row, 4 column matrix with values ranging from 1 to 36.

```
m[2,3]
```

```
## [1] 20
```

Retrieves the value in 2nd row and 3rd column i.e value 20.

```r
m[,3]
```

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

Retrieves all values which are present in 3rd column.

```r
m[2,]
```

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

Retrieves all values which are present in 2nd row.

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

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

cbind combines vectors column-wise and retrieves all values which are present in 3rd column and.

```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
```

Displays all rows and columns except the 3rd column.

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

```
##      [,1] [,2] [,3]
## [1,]   10   19   28
## [2,]   11   20   29
## [3,]   18   27   36
```

Displays all the rows, except rows ranging from 3 to 8(inclusive) and columns values ranging from 2 to 4(inclusive).

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
```

Cbind combines two vectors x1 and x2 where the x1 column value is only 3 and x2 column is ranging from 4 to 1 and 2 to 5 taking length size 8 thus creating a matrix[8,2].

```
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
```

Dimnames changes the names of the matrix dimensions. Since, x[[1]] signifies the row labels, letters ranging from 1 to 8 indexes are assigned to each of the row.

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

```
## x1 x2
##  3  3
```

apply function returns a vector after applying mean function on both the columns since the second parameter is given as 2 representating the function is being applied column-wise and trim function is applied to the resultant vector displaying 2 decimal places .

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

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

apply function returns a vector after applying sum function on both the columns of data x, since the second parameter is given as 2 representating the function is being applied column-wise and assigned to the variable col.sums.

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

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

apply function returns a vector after applying sum function on all the rows of data x, since the second parameter is given as 1 representating the function is being applied row-wise and assigned to the variable row.sums.

```
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
```

apply function returns a vector after applying sort function(sorting the resultant vector in ascending order by default) on all the columns of data x, since the second parameter is given as 2 representating the function is being applied column-wise

Question 2

```
# #install.packages("datasets")
# install.packages("dplyr") # useful in manipulating data
# install.packages("ggplot2") # useful for visualizations
# install.packages("lubridate")
# install.packages("rmarkdown")
#tinytex::install_tinytex()


###### Importing all the necessary libraries

library("datasets")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library("ggplot2")
library("lubridate")
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library("rmarkdown")
#install.packages("tinytex")
```

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.

```r
x<-15
y<-c(1,2,3,10,100)
z<-x*y
z
```

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

```r
sum(z)
```

```
## [1] 1740
```

2 b) Generate a sequence from 0 to 10 and a sequence from 5 to -5

```r
seq(0,10)
```

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

```r
seq(5,-5)
```

```
##  [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.

```r
seq(-3, 3, by = 0.1)
```

```
##  [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")
m <-c(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: 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)
Bmidata<-data.frame(age,sex,height,weight)
Bmidata
```

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

```
max(Bmidata$age)
```

```
## [1] 829
```

```
min(Bmidata$age)
```

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

```
library(dplyr)

selection <- (Bmidata[, 1] < 20) | (Bmidata[, 1] > 80) #all row , 1st column
selection
```

```
## [1] FALSE FALSE  TRUE  TRUE
```

```r
Bmidata[, 1][selection == TRUE] <- NA
Bmidata
```

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

```r
BMI <- round(as.numeric(Bmidata[, 4])/(as.numeric(Bmidata[, 3])/100)^2)
BMI
```

```
## [1] 21 19 26 22
```

```r
Bmidata<-data.frame(cbind(Bmidata,BMI))
Bmidata
```

```
##   age sex height weight BMI
## 1  21   m    181     69  21
## 2  35   f    173     58  19
## 3  NA   m    171     75  26
## 4  NA   e    166     60  22
```

Question 3

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.

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

```
##  [1]   9   8  12   6   1  10  10  10   8 516   8   6   4  19 100
```

3 (a) Compute the mean of x

```r
mean(x,trim = 0)
```

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

3 (b) Compute the standard deviation of x.

```r
sd(x)
```

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

3 (c) Compute the range of x

```r
range(x,na.rm = TRUE)
```

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

3 (d) Provide the five number summary of x

```r
fivenum(x)
```

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

From the output we get the following values:

The minimum: 1

The first quartile: 7

The median: 9

The third quartile: 11

The maximum: 516

3 (e) Is there any NA in x?

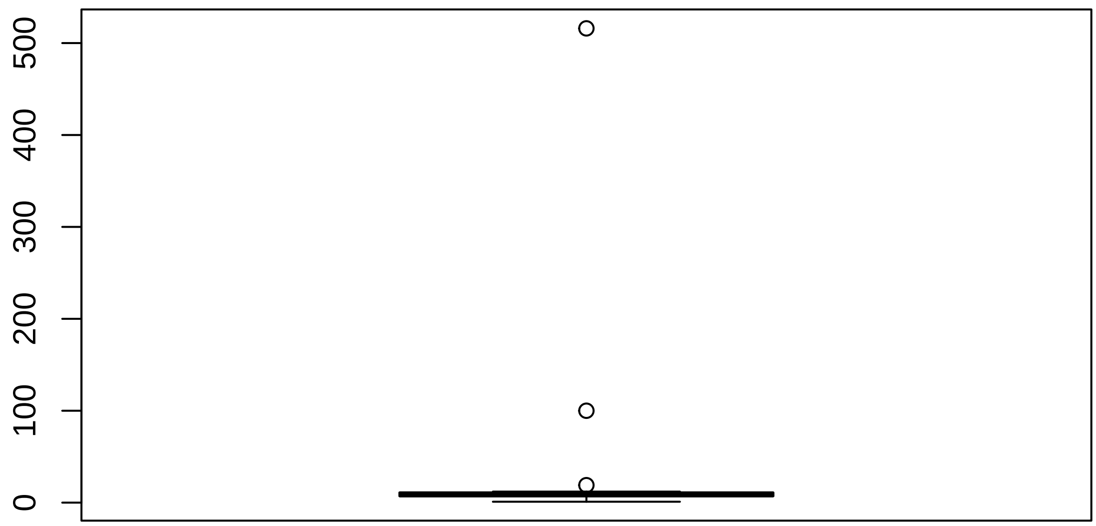
```r
is.na(x)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE
```

The output shows that there is no NA present.

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

```r
boxplot(x)
```



Yes, there are outliers present which is shown by the output. ###### To remove the outlier, we run the below R code

```
x_remove_out <- x[!x %in% boxplot.stats(x)$out]
x_remove_out
```

```
##  [1]  9  8 12  6  1 10 10 10  8  8  6  4
```

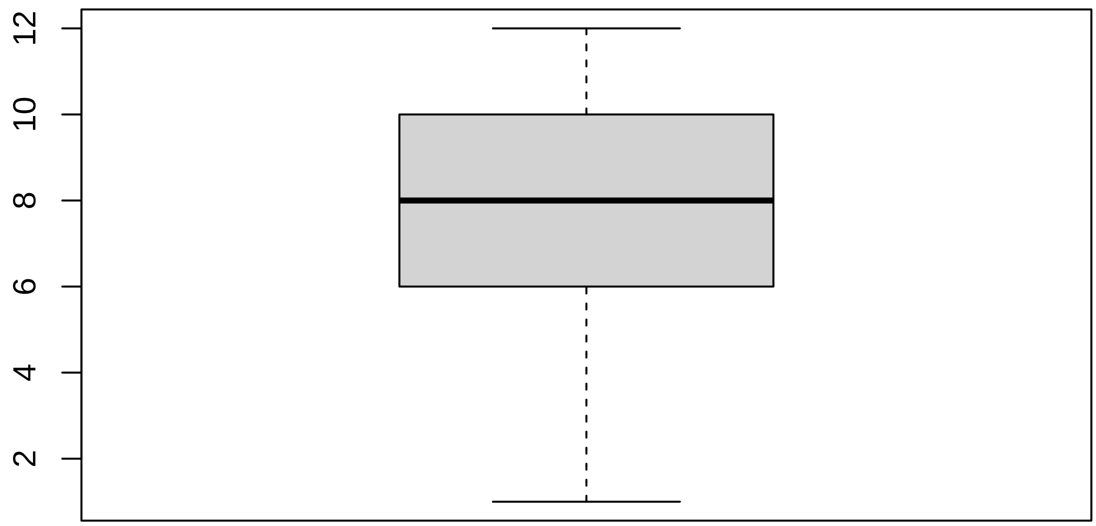
We have removed 3 values from our data. We have got this by below code

```
length(x) - length(x_remove_out)
```

```
## [1] 3
```

The below boxplot is displayed by removing outliers.

```
boxplot(x_remove_out)
```



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.

```
library(lubridate)
data <- read.csv("arbuthnot.csv", header = T)
data
```

```
##     X year boys girls
## 1   1 1629 5218  4683
## 2   2 1630 4858  4457
## 3   3 1631 4422  4102
## 4   4 1632 4994  4590
## 5   5 1633 5158  4839
## 6   6 1634 5035  4820
## 7   7 1635 5106  4928
## 8   8 1636 4917  4605
## 9   9 1637 4703  4457
## 10 10 1638 5359  4952
## 11 11 1639 5366  4784
## 12 12 1640 5518  5332
## 13 13 1641 5470  5200
## 14 14 1642 5460  4910
## 15 15 1643 4793  4617
## 16 16 1644 4107  3997
```

```
## 17 17 1645 4047  3919
## 18 18 1646 3768  3395
## 19 19 1647 3796  3536
## 20 20 1648 3363  3181
## 21 21 1649 3079  2746
## 22 22 1650 2890  2722
## 23 23 1651 3231  2840
## 24 24 1652 3220  2908
## 25 25 1653 3196  2959
## 26 26 1654 3441  3179
## 27 27 1655 3655  3349
## 28 28 1656 3668  3382
## 29 29 1657 3396  3289
## 30 30 1658 3157  3013
## 31 31 1659 3209  2781
## 32 32 1660 3724  3247
## 33 33 1661 4748  4107
## 34 34 1662 5216  4803
## 35 35 1663 5411  4881
## 36 36 1664 6041  5681
## 37 37 1665 5114  4858
## 38 38 1666 4678  4319
## 39 39 1667 5616  5322
## 40 40 1668 6073  5560
## 41 41 1669 6506  5829
## 42 42 1670 6278  5719
## 43 43 1671 6449  6061
## 44 44 1672 6443  6120
## 45 45 1673 6073  5822
## 46 46 1674 6113  5738
## 47 47 1675 6058  5717
## 48 48 1676 6552  5847
## 49 49 1677 6423  6203
## 50 50 1678 6568  6033
## 51 51 1679 6247  6041
## 52 52 1680 6548  6299
## 53 53 1681 6822  6533
## 54 54 1682 6909  6744
## 55 55 1683 7577  7158
## 56 56 1684 7575  7127
## 57 57 1685 7484  7246
## 58 58 1686 7575  7119
## 59 59 1687 7737  7214
## 60 60 1688 7487  7101
## 61 61 1689 7604  7167
## 62 62 1690 7909  7302
## 63 63 1691 7662  7392
## 64 64 1692 7602  7316
## 65 65 1693 7676  7483
## 66 66 1694 6985  6647
## 67 67 1695 7263  6713
## 68 68 1696 7632  7229
## 69 69 1697 8062  7767
## 70 70 1698 8426  7626
```

```
## 71 71 1699 7911  7452
## 72 72 1700 7578  7061
## 73 73 1701 8102  7514
## 74 74 1702 8031  7656
## 75 75 1703 7765  7683
## 76 76 1704 6113  5738
## 77 77 1705 8366  7779
## 78 78 1706 7952  7417
## 79 79 1707 8379  7687
## 80 80 1708 8239  7623
## 81 81 1709 7840  7380
## 82 82 1710 7640  7288
```

4 a) What is the dimension of this dataset?

```r
dim(data)
```

```
## [1] 82  4
```

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

```r
ls(data)
```

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

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

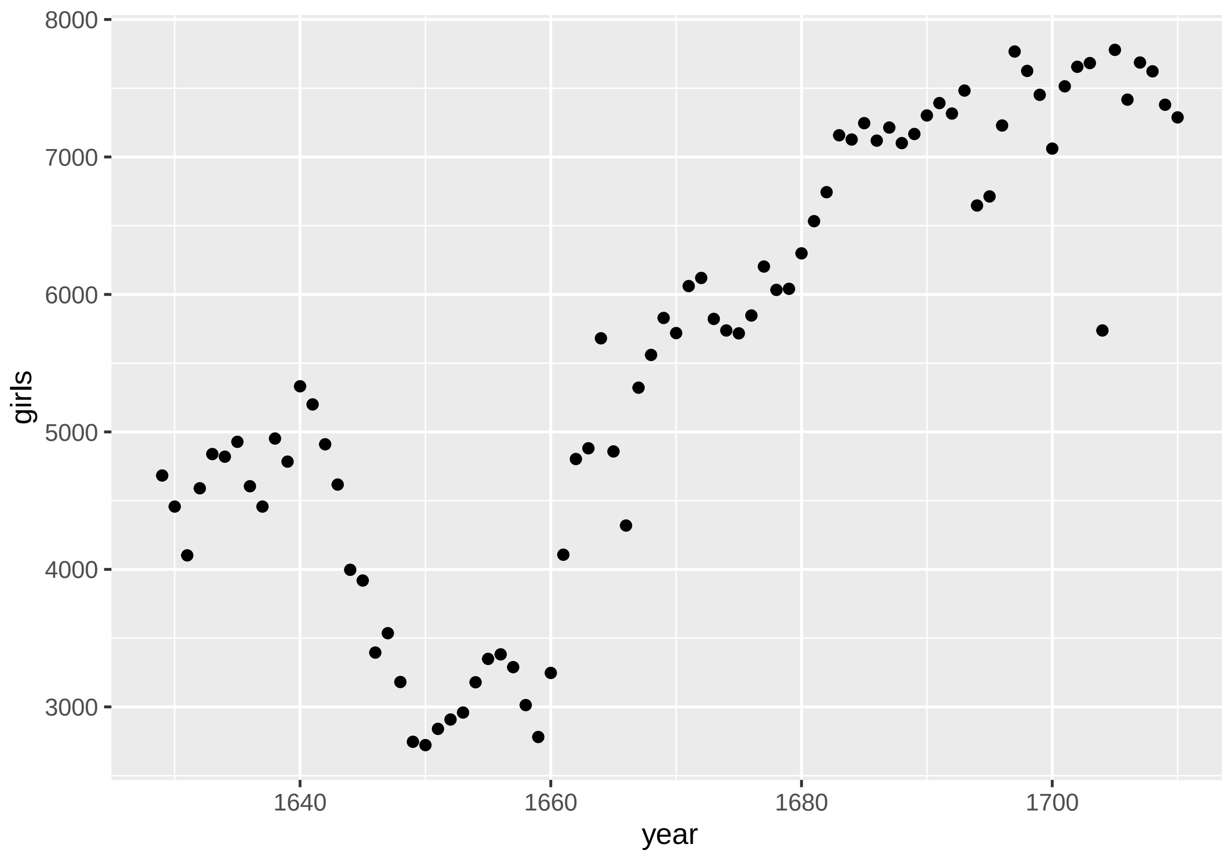
```r
data$girls
```

```
##  [1] 4683 4457 4102 4590 4839 4820 4928 4605 4457 4952 4784 5332 5200 4910 4617
## [16] 3997 3919 3395 3536 3181 2746 2722 2840 2908 2959 3179 3349 3382 3289 3013
## [31] 2781 3247 4107 4803 4881 5681 4858 4319 5322 5560 5829 5719 6061 6120 5822
## [46] 5738 5717 5847 6203 6033 6041 6299 6533 6744 7158 7127 7246 7119 7214 7101
## [61] 7167 7302 7392 7316 7483 6647 6713 7229 7767 7626 7452 7061 7514 7656 7683
## [76] 5738 7779 7417 7687 7623 7380 7288
```

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

There is initially an increase in the number of girls baptised, which peaks around 1640. After 1640 there is a decrease in the number of girls baptised but the number begins to increase again in 1660. Overall the trend is an increase in the number of girls baptised

```r
library(ggplot2)
ggplot(data, aes(x = year, y = girls)) +
  geom_point()
```



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

To calculate proportion of boys over girls

```
data
```

```
##     X year boys girls
## 1   1 1629 5218  4683
## 2   2 1630 4858  4457
## 3   3 1631 4422  4102
## 4   4 1632 4994  4590
## 5   5 1633 5158  4839
## 6   6 1634 5035  4820
## 7   7 1635 5106  4928
## 8   8 1636 4917  4605
## 9   9 1637 4703  4457
## 10 10 1638 5359  4952
## 11 11 1639 5366  4784
## 12 12 1640 5518  5332
## 13 13 1641 5470  5200
## 14 14 1642 5460  4910
## 15 15 1643 4793  4617
## 16 16 1644 4107  3997
## 17 17 1645 4047  3919
## 18 18 1646 3768  3395
## 19 19 1647 3796  3536
## 20 20 1648 3363  3181
```

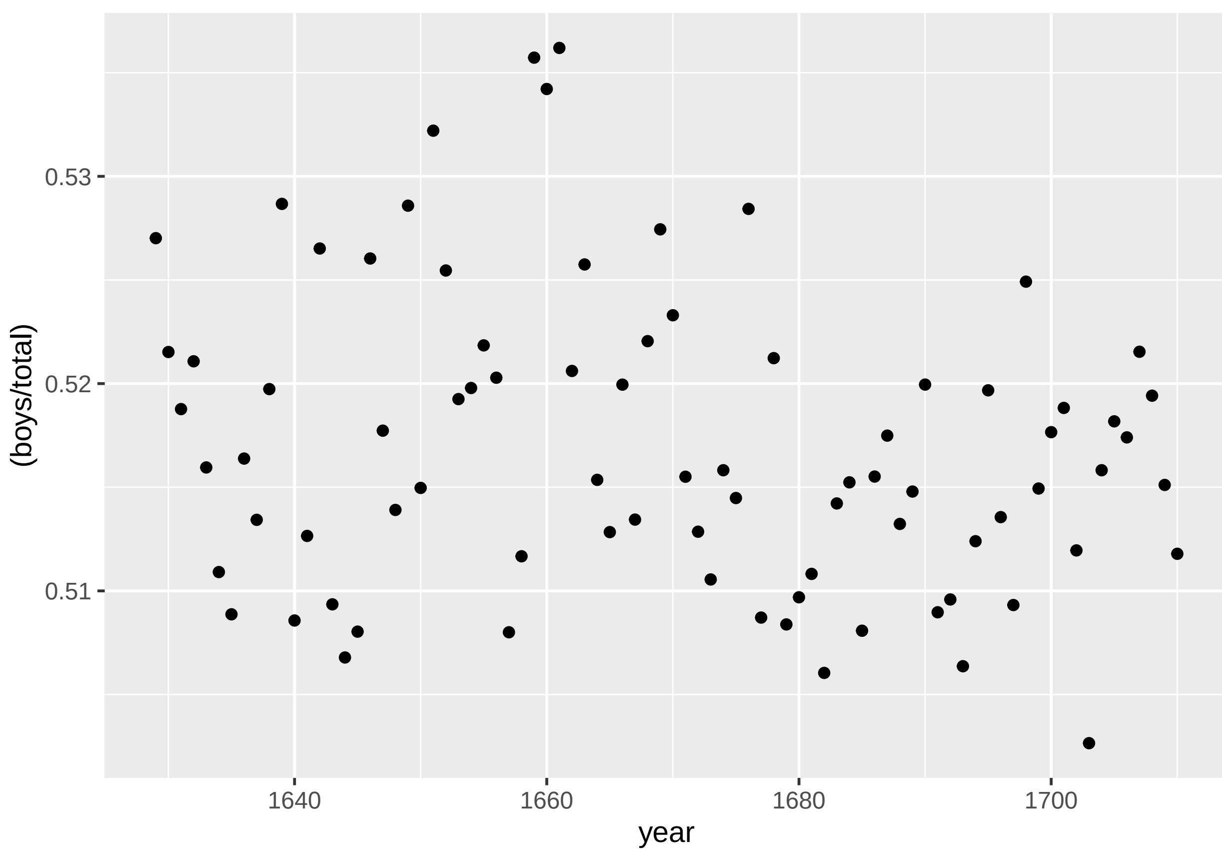
```
## 21 21 1649 3079  2746
## 22 22 1650 2890  2722
## 23 23 1651 3231  2840
## 24 24 1652 3220  2908
## 25 25 1653 3196  2959
## 26 26 1654 3441  3179
## 27 27 1655 3655  3349
## 28 28 1656 3668  3382
## 29 29 1657 3396  3289
## 30 30 1658 3157  3013
## 31 31 1659 3209  2781
## 32 32 1660 3724  3247
## 33 33 1661 4748  4107
## 34 34 1662 5216  4803
## 35 35 1663 5411  4881
## 36 36 1664 6041  5681
## 37 37 1665 5114  4858
## 38 38 1666 4678  4319
## 39 39 1667 5616  5322
## 40 40 1668 6073  5560
## 41 41 1669 6506  5829
## 42 42 1670 6278  5719
## 43 43 1671 6449  6061
## 44 44 1672 6443  6120
## 45 45 1673 6073  5822
## 46 46 1674 6113  5738
## 47 47 1675 6058  5717
## 48 48 1676 6552  5847
## 49 49 1677 6423  6203
## 50 50 1678 6568  6033
## 51 51 1679 6247  6041
## 52 52 1680 6548  6299
## 53 53 1681 6822  6533
## 54 54 1682 6909  6744
## 55 55 1683 7577  7158
## 56 56 1684 7575  7127
## 57 57 1685 7484  7246
## 58 58 1686 7575  7119
## 59 59 1687 7737  7214
## 60 60 1688 7487  7101
## 61 61 1689 7604  7167
## 62 62 1690 7909  7302
## 63 63 1691 7662  7392
## 64 64 1692 7602  7316
## 65 65 1693 7676  7483
## 66 66 1694 6985  6647
## 67 67 1695 7263  6713
## 68 68 1696 7632  7229
## 69 69 1697 8062  7767
## 70 70 1698 8426  7626
## 71 71 1699 7911  7452
## 72 72 1700 7578  7061
## 73 73 1701 8102  7514
## 74 74 1702 8031  7656
```

```
## 75 75 1703 7765  7683
## 76 76 1704 6113  5738
## 77 77 1705 8366  7779
## 78 78 1706 7952  7417
## 79 79 1707 8379  7687
## 80 80 1708 8239  7623
## 81 81 1709 7840  7380
## 82 82 1710 7640  7288
```

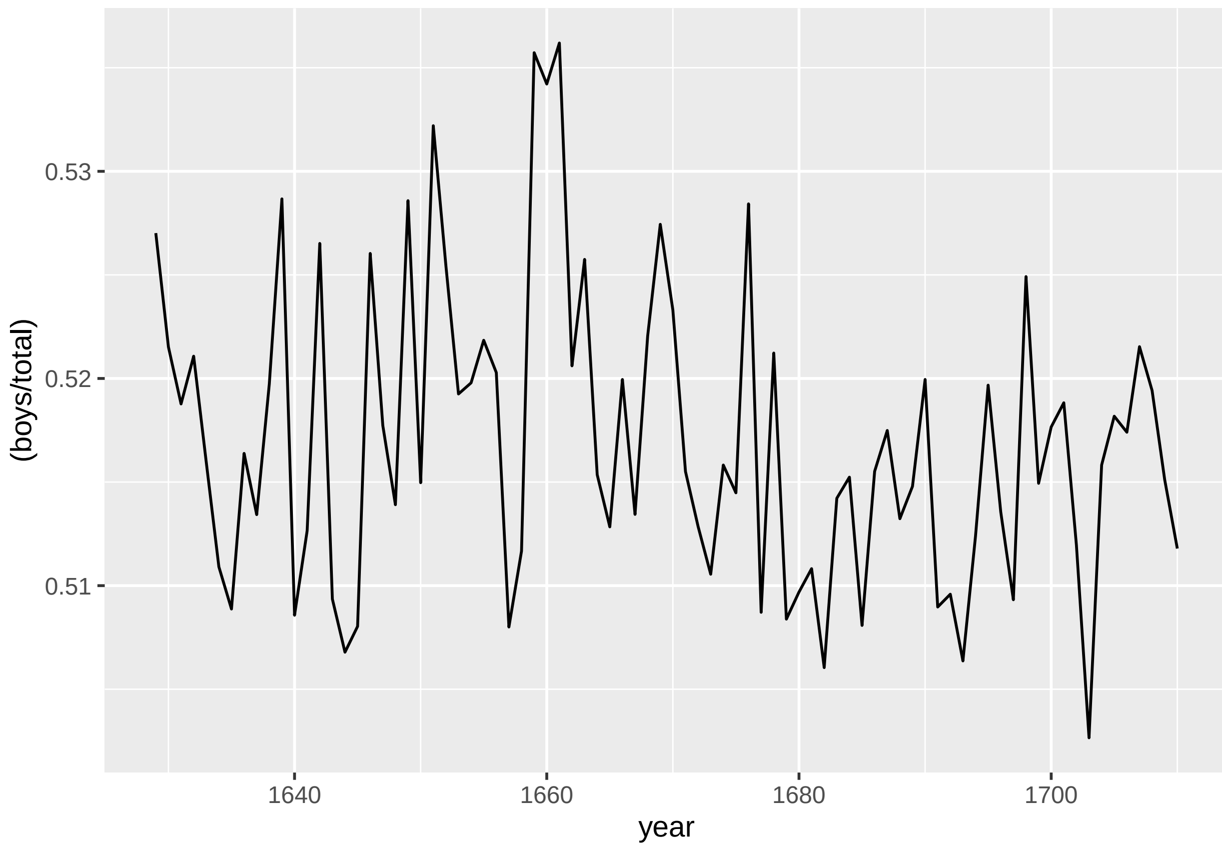
```
total<- (data$boys+data$girls)
total
```

```
##  [1]  9901  9315  8524  9584  9997  9855 10034  9522  9160 10311 10150 10850
## [13] 10670 10370  9410  8104  7966  7163  7332  6544  5825  5612  6071  6128
## [25]  6155  6620  7004  7050  6685  6170  5990  6971  8855 10019 10292 11722
## [37]  9972  8997 10938 11633 12335 11997 12510 12563 11895 11851 11775 12399
## [49] 12626 12601 12288 12847 13355 13653 14735 14702 14730 14694 14951 14588
## [61] 14771 15211 15054 14918 15159 13632 13976 14861 15829 16052 15363 14639
## [73] 15616 15687 15448 11851 16145 15369 16066 15862 15220 14928
```

```
#in a Point
ggplot(data, aes(x = year, y = (boys/total))) +
  geom_point()
```



```r
ggplot(data, aes(x = year, y = (boys/total))) +
  geom_line()
```




```r
library(dplyr)
data %>% group_by(boys) %>% summarize(f = n())
```

```
## # A tibble: 79 x 2
##     boys     f
##    <int> <int>
##  1  2890     1
##  2  3079     1
##  3  3157     1
##  4  3196     1
##  5  3209     1
##  6  3220     1
##  7  3231     1
##  8  3363     1
##  9  3396     1
## 10  3441     1
## # ... with 69 more rows
```

Plot of the proportion of boys over time. What do you see

We can see from the mutate function which provides comparison and appends the variable in dataframe

```r
data <- data %>% mutate(comparison_if_more_boys = boys > girls)
data
```

```
##     X year boys girls comparison_if_more_boys
## 1   1 1629 5218  4683                    TRUE
## 2   2 1630 4858  4457                    TRUE
## 3   3 1631 4422  4102                    TRUE
## 4   4 1632 4994  4590                    TRUE
## 5   5 1633 5158  4839                    TRUE
## 6   6 1634 5035  4820                    TRUE
## 7   7 1635 5106  4928                    TRUE
## 8   8 1636 4917  4605                    TRUE
## 9   9 1637 4703  4457                    TRUE
## 10 10 1638 5359  4952                    TRUE
## 11 11 1639 5366  4784                    TRUE
## 12 12 1640 5518  5332                    TRUE
## 13 13 1641 5470  5200                    TRUE
## 14 14 1642 5460  4910                    TRUE
## 15 15 1643 4793  4617                    TRUE
## 16 16 1644 4107  3997                    TRUE
## 17 17 1645 4047  3919                    TRUE
## 18 18 1646 3768  3395                    TRUE
## 19 19 1647 3796  3536                    TRUE
## 20 20 1648 3363  3181                    TRUE
## 21 21 1649 3079  2746                    TRUE
## 22 22 1650 2890  2722                    TRUE
## 23 23 1651 3231  2840                    TRUE
## 24 24 1652 3220  2908                    TRUE
## 25 25 1653 3196  2959                    TRUE
## 26 26 1654 3441  3179                    TRUE
## 27 27 1655 3655  3349                    TRUE
## 28 28 1656 3668  3382                    TRUE
## 29 29 1657 3396  3289                    TRUE
## 30 30 1658 3157  3013                    TRUE
## 31 31 1659 3209  2781                    TRUE
## 32 32 1660 3724  3247                    TRUE
## 33 33 1661 4748  4107                    TRUE
## 34 34 1662 5216  4803                    TRUE
## 35 35 1663 5411  4881                    TRUE
## 36 36 1664 6041  5681                    TRUE
## 37 37 1665 5114  4858                    TRUE
## 38 38 1666 4678  4319                    TRUE
## 39 39 1667 5616  5322                    TRUE
## 40 40 1668 6073  5560                    TRUE
## 41 41 1669 6506  5829                    TRUE
## 42 42 1670 6278  5719                    TRUE
## 43 43 1671 6449  6061                    TRUE
## 44 44 1672 6443  6120                    TRUE
## 45 45 1673 6073  5822                    TRUE
## 46 46 1674 6113  5738                    TRUE
## 47 47 1675 6058  5717                    TRUE
## 48 48 1676 6552  5847                    TRUE
## 49 49 1677 6423  6203                    TRUE
```

```
## 50 50 1678 6568  6033                    TRUE
## 51 51 1679 6247  6041                    TRUE
## 52 52 1680 6548  6299                    TRUE
## 53 53 1681 6822  6533                    TRUE
## 54 54 1682 6909  6744                    TRUE
## 55 55 1683 7577  7158                    TRUE
## 56 56 1684 7575  7127                    TRUE
## 57 57 1685 7484  7246                    TRUE
## 58 58 1686 7575  7119                    TRUE
## 59 59 1687 7737  7214                    TRUE
## 60 60 1688 7487  7101                    TRUE
## 61 61 1689 7604  7167                    TRUE
## 62 62 1690 7909  7302                    TRUE
## 63 63 1691 7662  7392                    TRUE
## 64 64 1692 7602  7316                    TRUE
## 65 65 1693 7676  7483                    TRUE
## 66 66 1694 6985  6647                    TRUE
## 67 67 1695 7263  6713                    TRUE
## 68 68 1696 7632  7229                    TRUE
## 69 69 1697 8062  7767                    TRUE
## 70 70 1698 8426  7626                    TRUE
## 71 71 1699 7911  7452                    TRUE
## 72 72 1700 7578  7061                    TRUE
## 73 73 1701 8102  7514                    TRUE
## 74 74 1702 8031  7656                    TRUE
## 75 75 1703 7765  7683                    TRUE
## 76 76 1704 6113  5738                    TRUE
## 77 77 1705 8366  7779                    TRUE
## 78 78 1706 7952  7417                    TRUE
## 79 79 1707 8379  7687                    TRUE
## 80 80 1708 8239  7623                    TRUE
## 81 81 1709 7840  7380                    TRUE
## 82 82 1710 7640  7288                    TRUE
```

######We can see that there are more boys than the girls on year on basis.

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

```
data$total<-data$boys+data$girls
 data[data$total== max(data$total),"year"]
```

```
## [1] 1705
```

Question 5

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.

```
summary(attitude)
```

```
##      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?There are 30 observations in the attitude dataset. We used "nrow(attitude)" to display the number of observations.

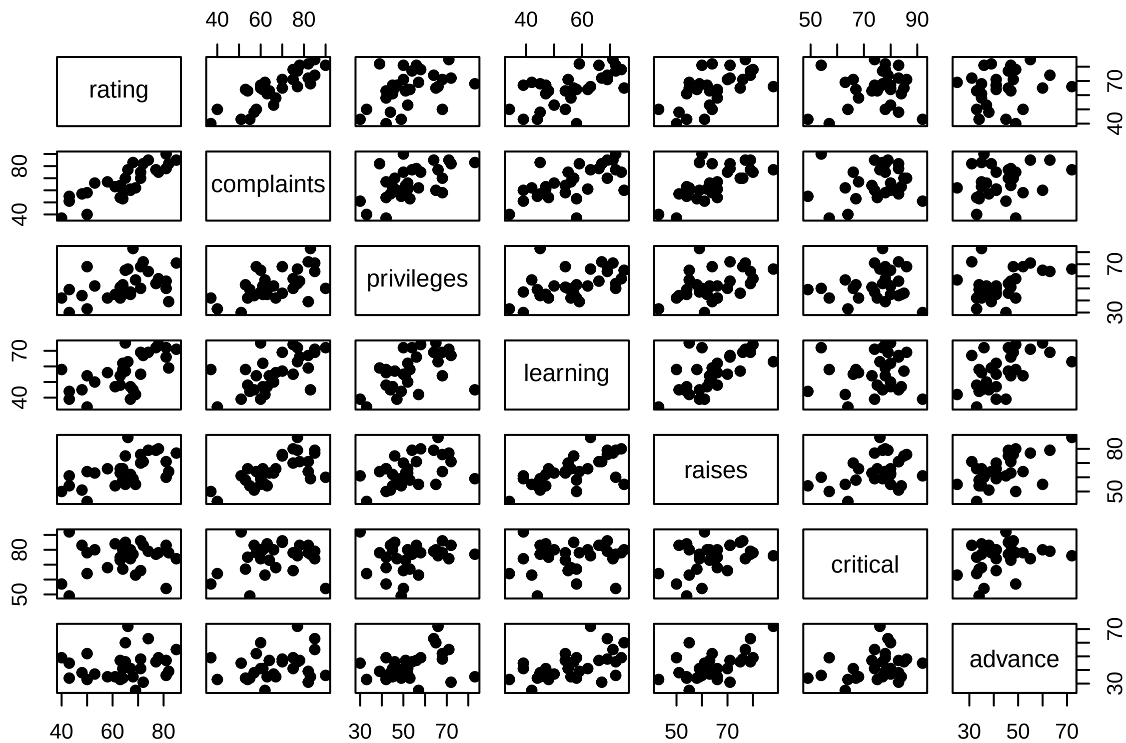
```
nrow(attitude)
```

```
## [1] 30
```

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

ANS) The scatterplot below shows that the rating is highly correlated with complaints. As the rating are increasing the complaints are increasing.

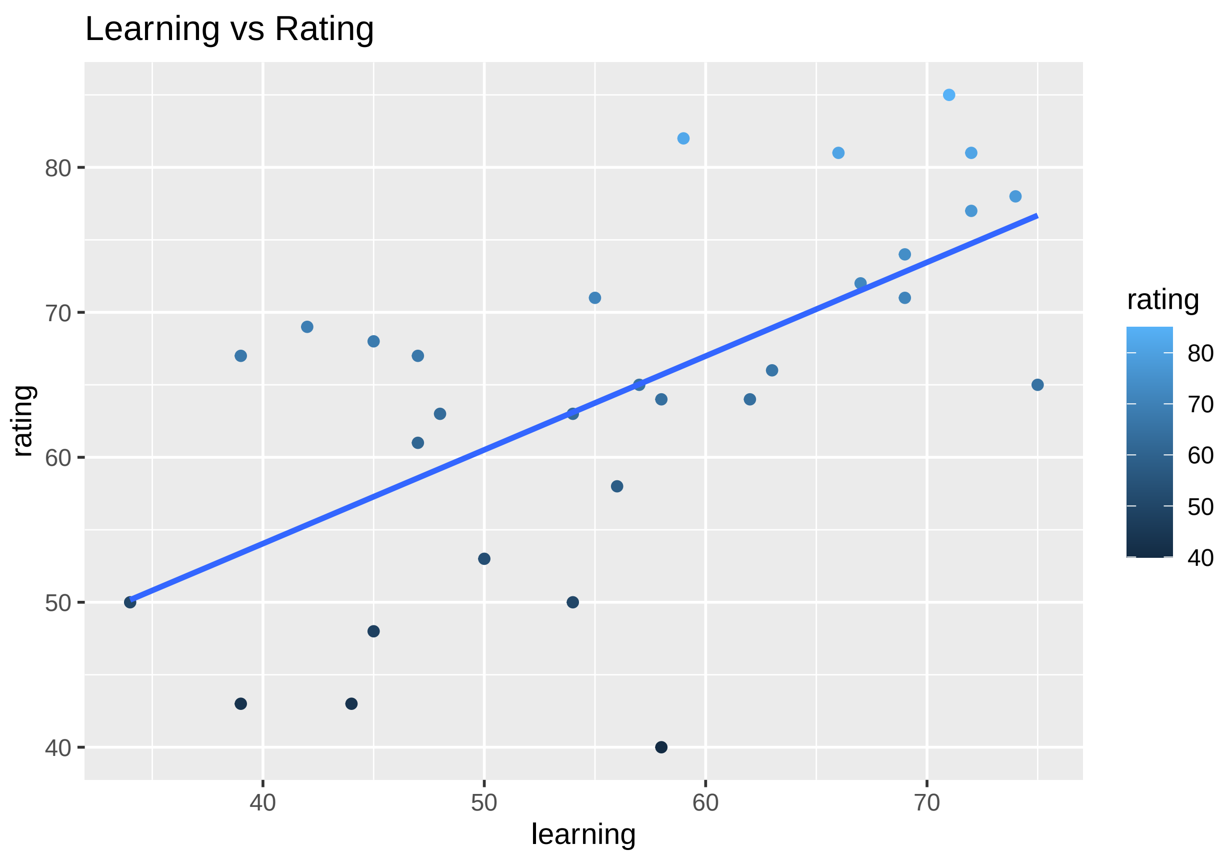
```
pairs(attitude[,1:7], pch = 19)
```



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

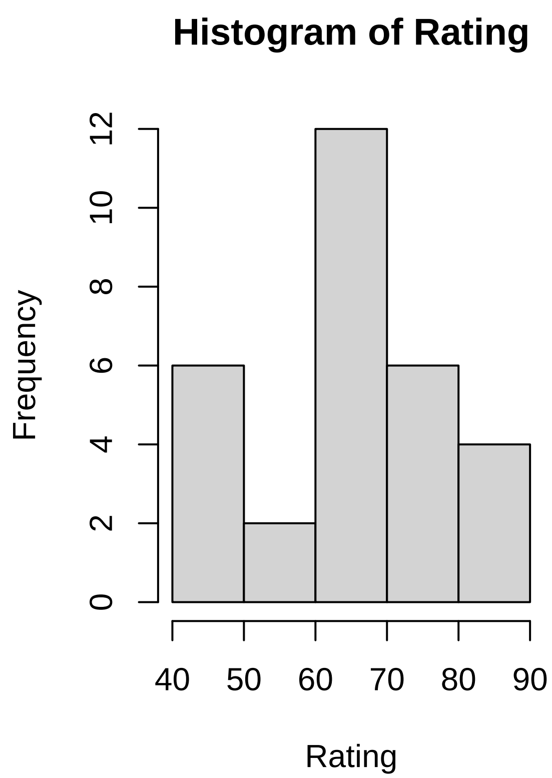
```
library(ggplot2)
ggplot(attitude, aes(x=learning, y=rating, color=rating)) + ggtitle("Learning vs Rating")+
  geom_point() +
  geom_smooth(method=lm, se=FALSE)
```

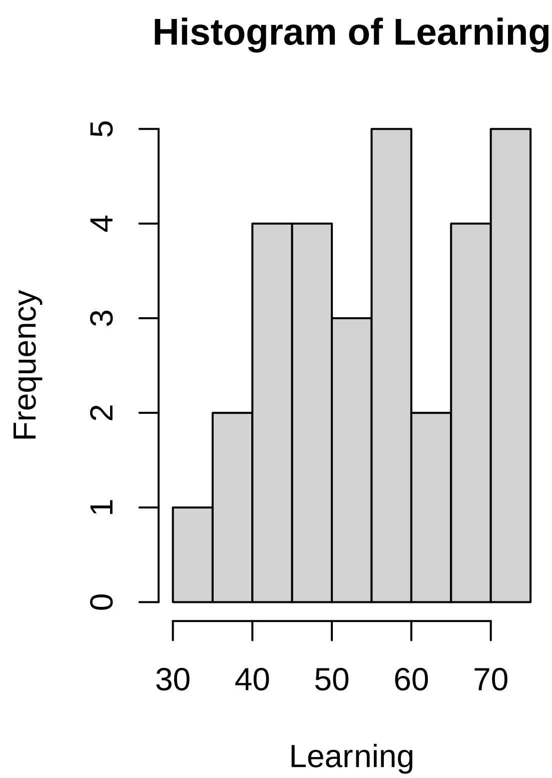
```
## `geom_smooth()` using formula 'y ~ x'
```



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.

```
par(mfrow=c(1,2))
hist(attitude$rating,xlab="Rating",main = "Histogram of Rating")
hist(attitude$learning,xlab="Learning",main = "Histogram of Learning")
```





Question 6

6) To answer this question use the R built-in dataset "mtcars"

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 <- mtcars
?mtcars
```

```
## starting httpd help server ... done
```

```
cars_data <- mtcars
cars_data
```

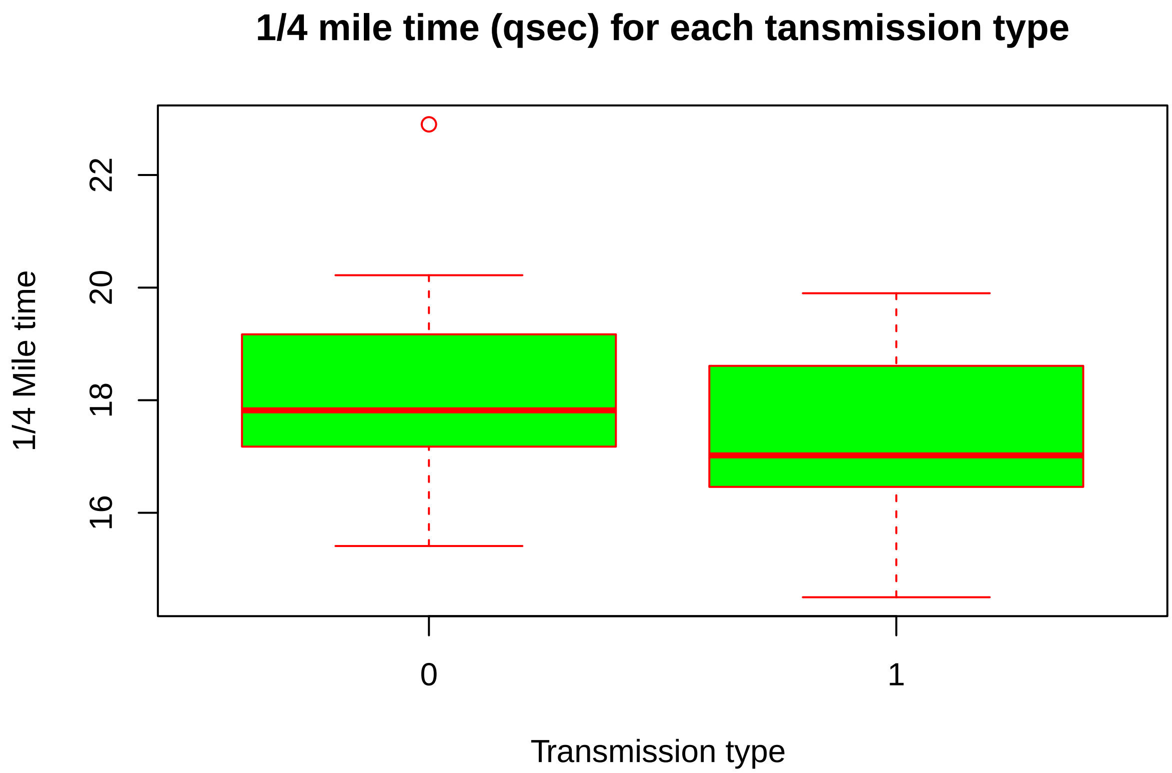
```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
```

```
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

ANS) The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

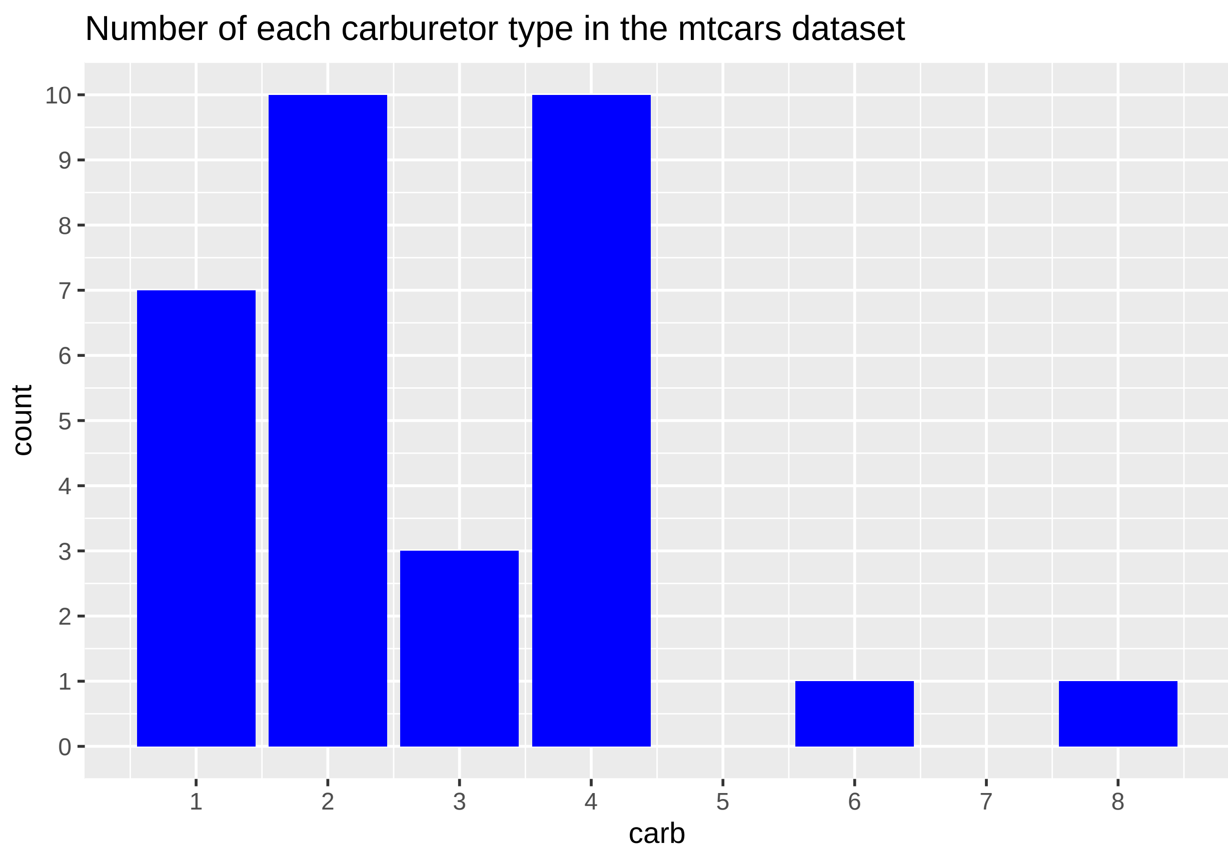
6 b)Create a box plot using ggplot showing the range of values of 1/4 mile time (qsec) for each tansmission type (am, 0 = automatic, 1 = manual) from the mtcars data set. Use "Transmission Type" and "1/4 Mile Time" for your y- and x-axes respectively. Also, add the title to your graph.

```
library(ggplot2)
data(mtcars)
boxplot(qsec~am,
        data=mtcars,
        main="1/4 mile time (qsec) for each tansmission type",
        xlab="Transmission type ",
        ylab="1/4 Mile time",
        col="green",
        border="red" )
```



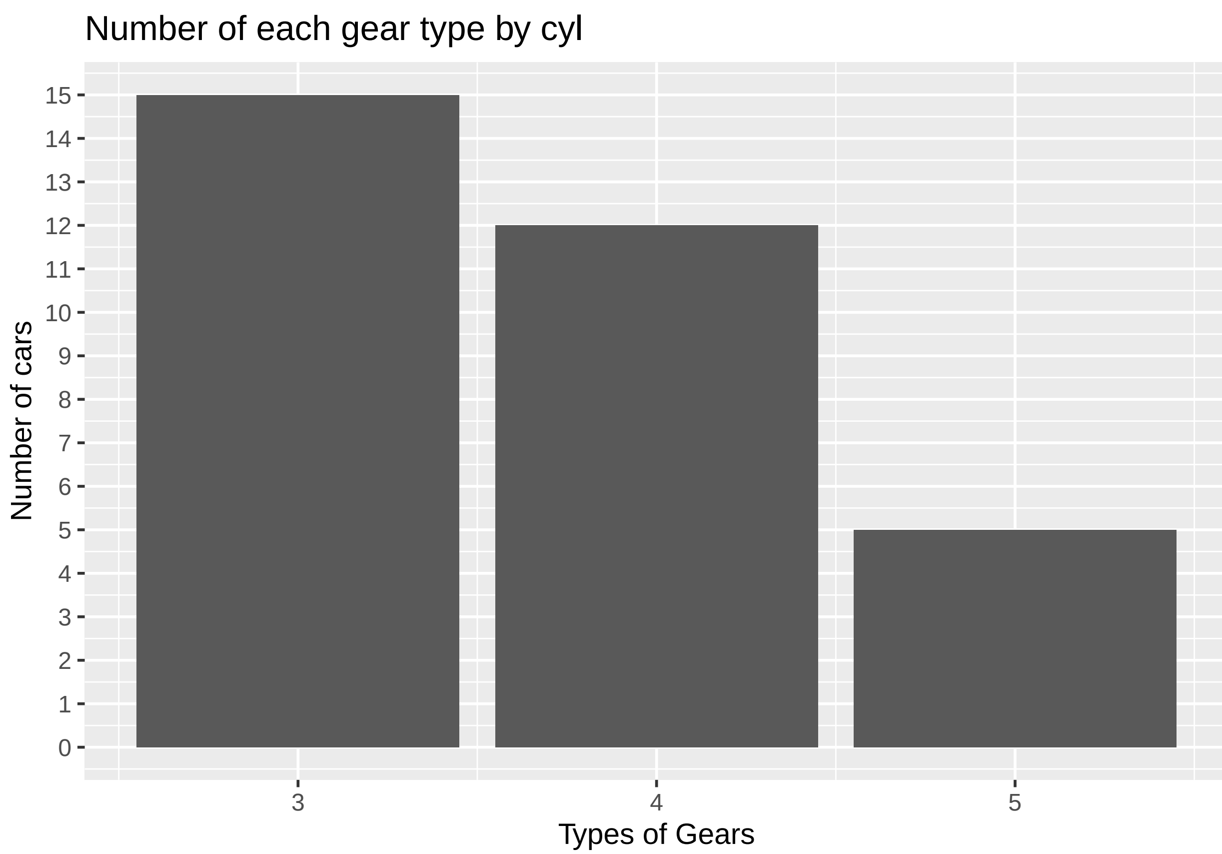
6 (c) Create a bar graph using ggplot, that shows the number of each carb type in mtcars.

```
library(ggplot2)
Bar_graph<-ggplot(data=mtcars, aes(x=carb),) +
  ggtitle ("Number of each carburetor type in the mtcars dataset")+
  geom_bar(fill = "blue")+
  scale_x_continuous(breaks = seq(0,8, by = 1))+
scale_y_continuous(breaks = seq(0, 12, by = 1))
  Bar_graph
```



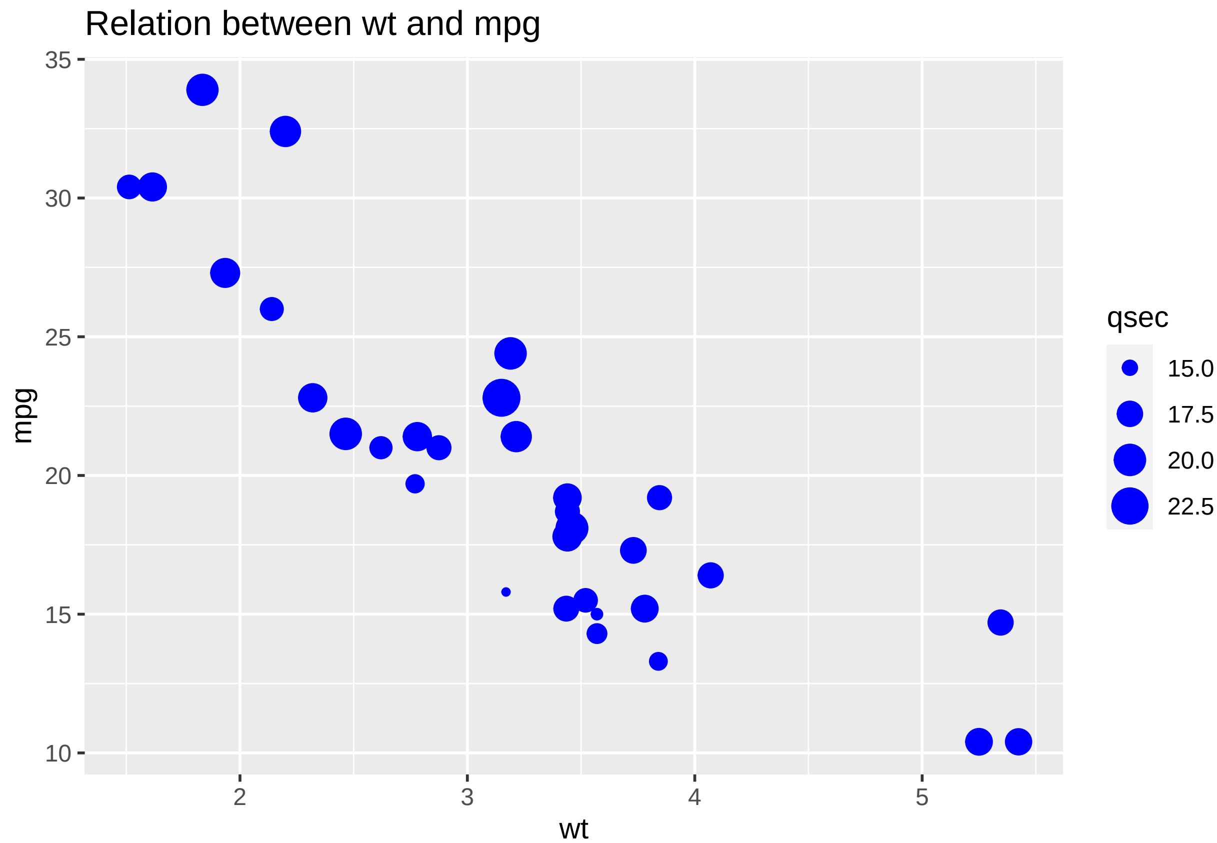
6 (d) Next show a stacked bar graph using ggplot of the number of each gear type and how they are further divided out by cyl. Add labels and a title to your plot. Anything dependent on a variable needs to go in aes(). Anything constant does not need aes

```
stacked_bar_graph<- ggplot(mtcars, aes( x=gear, fill=cyl )) +
  ggtitle ("Number of each gear type by cyl ") +
  geom_bar()+
  labs(x=" Types of Gears ", y="Number of cars") +
  scale_fill_discrete(name="Type of cyclinders")+
  scale_y_continuous(breaks = seq(0, 15, by = 1))
  stacked_bar_graph
```



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

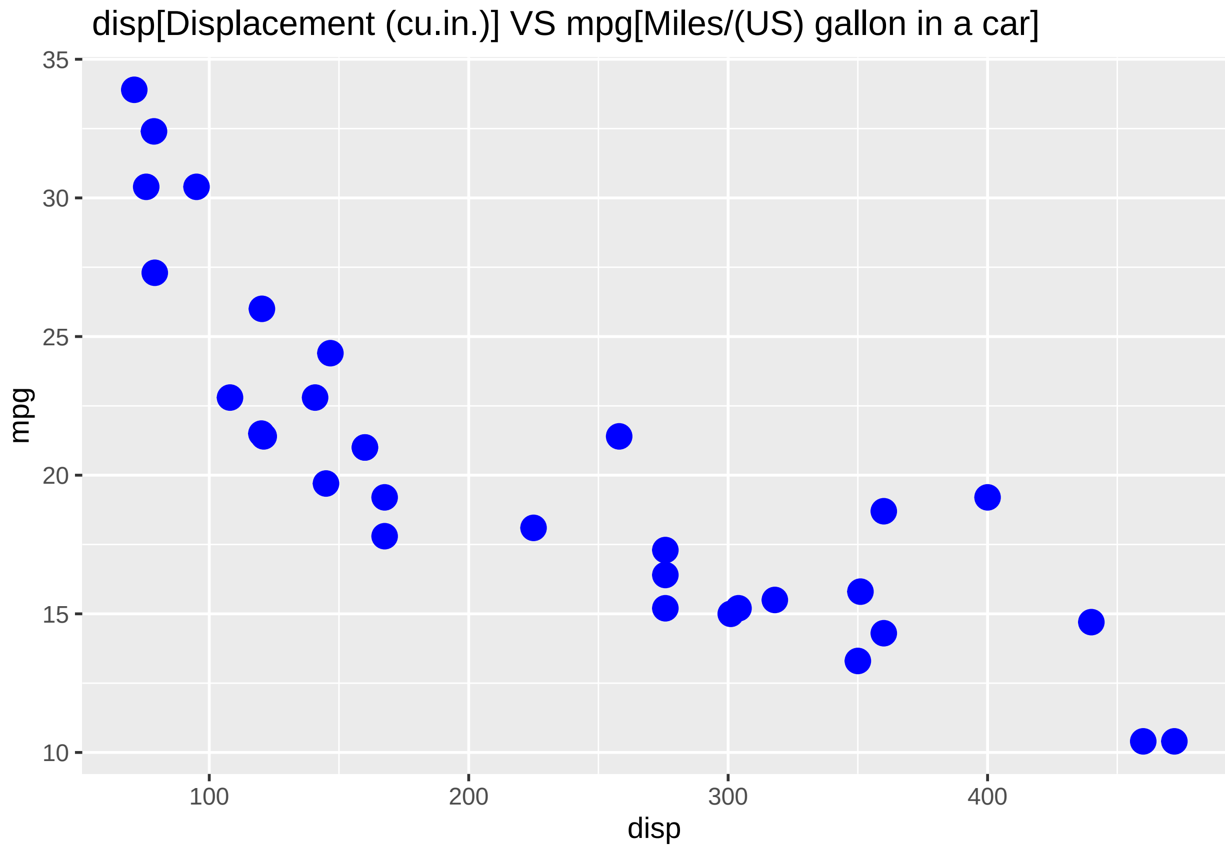
```
scatter_plot<- ggplot(mtcars, aes(x=wt, y=mpg)) +
  ggtitle ("Relation between wt and mpg ")+
  geom_point(aes(size=qsec),color="blue")
  scatter_plot
```



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

```
scatter_plot_relation<- ggplot(mtcars, aes(x=disp, y=mpg)) +
  ggtitle (" disp[Displacement (cu.in.)] VS mpg[Miles/(US) gallon in a car] ")+
  geom_point(size= 4,color="blue")

scatter_plot_relation
```

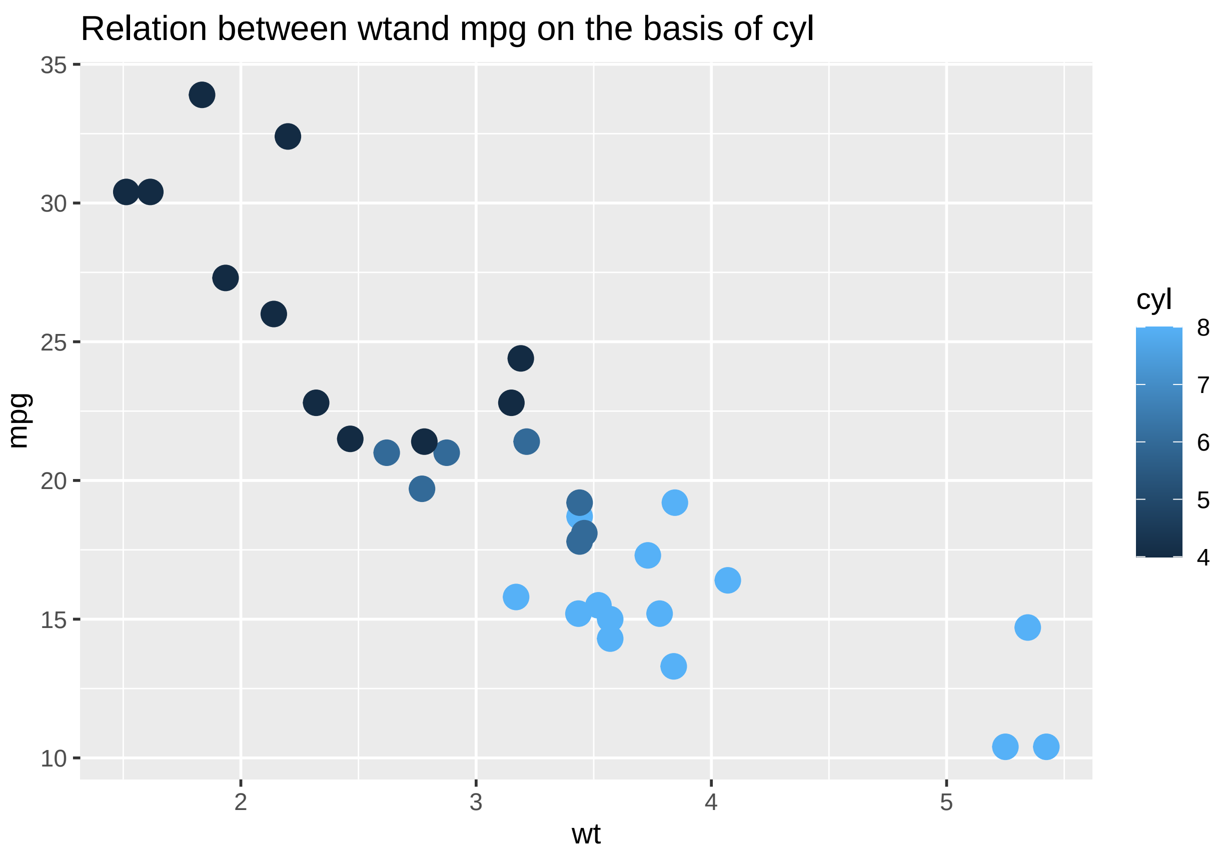



```
?mtcars
```

Ans) From the graph, it is clear that the relation between Displacement and Miles/(US) gallon is inversely proportional to each other . In the graph, for a car having disp of 100(cu.in.) covers more distance than a car having a disp parameter of more than 400(cu.in.).

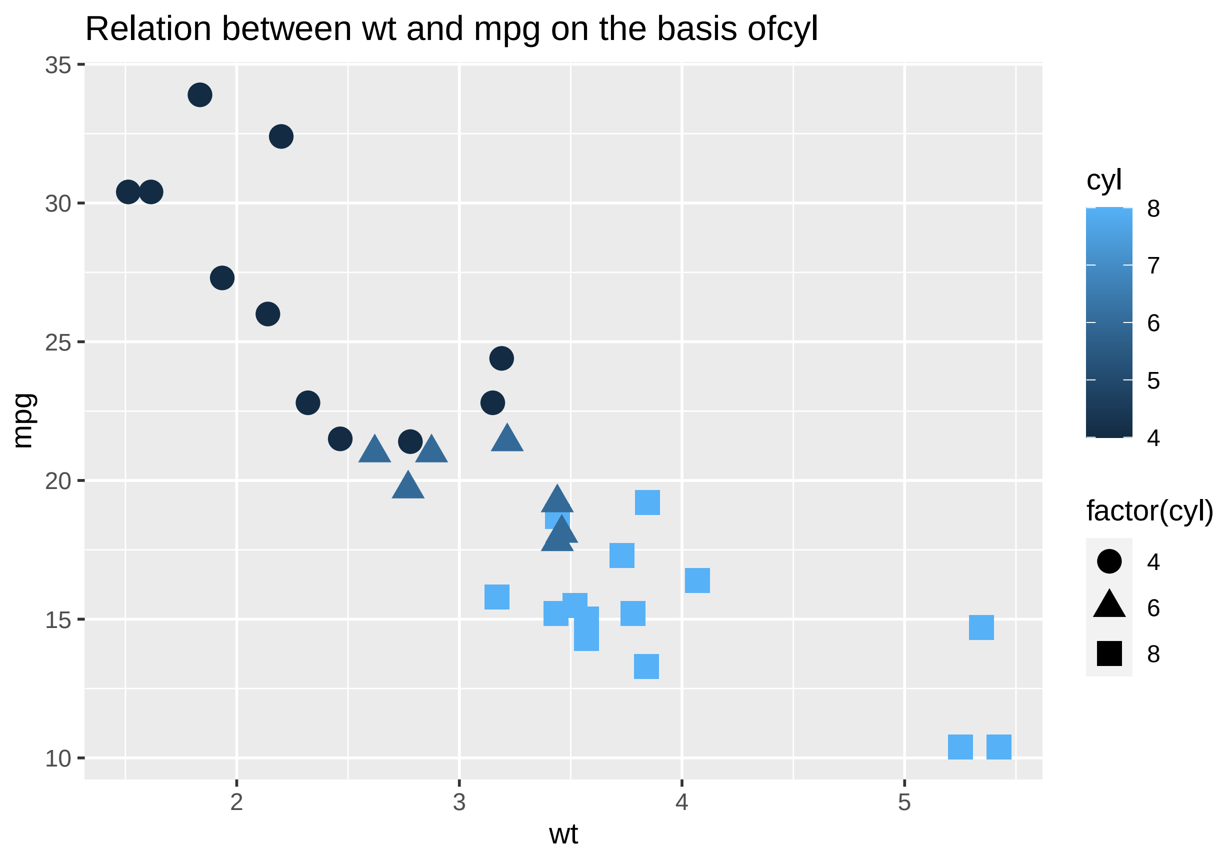
6 (g) Create a scatter plot that shows the relationship between various car weights (wt), miles per gallon (mpg) and engine cylinders (cyl). Use colored points to show the different cylinders in the plot. Note: you will need to convert cyl to a factor. You will need the function factor() to do this.

```
Scatter_plot_3 <- ggplot(mtcars, aes(x=wt, y=mpg, color=cyl, size=cyl)) +
  ggtitle ("Relation between wtand mpg on the basis of cyl ")+
  geom_point(size=4)
Scatter_plot_3
```



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

```
  Scatter_plot_4 <- ggplot(mtcars, aes(x=wt, y=mpg, color=cyl, size=cyl, shape=factor(cyl))) +
   ggtitle ("Relation between wt and mpg on the basis ofcyl ")+
   geom_point(size=4)
 Scatter_plot_4
```



Question 7

getwd() //setwd()

```
library(dplyr)
tinytex::install_tinytex()
```

Installing the necessary packages Reading the CSV file into a data frame

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

DataFrame - GAPMINDER

```
View(gm)
```

Question 7) Download the gapminder.csv data and read it into R. Assign the data to an object called gm. Use this data set to answer the following questions. Try to use the functions in dplyr package if possible.

```
gm %>% dplyr::group_by(continent) %>% dplyr::summarise(Distinct_Country_Count = n_distinct(country))
```

7(a) How many unique countries are represented per continent?

```
## # A tibble: 5 x 2
##   continent Distinct_Country_Count
##   <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 captia in 1997?

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

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

7 (c) According to the data available, what was the average life expectancy across each continent inthe 1980s?

```
gm %>% group_by(continent) %>% filter(year >= 1980 | year < 1990) %>% summarise(n = mean(lifeExp))
```

```
## # A tibble: 5 x 2
##   continent     n
##   <chr>     <dbl>
## 1 Africa     48.9
## 2 Americas   64.7
## 3 Asia       60.1
## 4 Europe     71.9
## 5 Oceania    74.3
```

7 (d): What 5 countries have the highest total GDP over all years combined? GDP =gdpPerCaptia * Population

```
gm %>% dplyr:: mutate(Total_GDP = gdpPercap * pop)%>% dplyr::group_by(country)%>% summarise(Total_GDP =
```

```
## # 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
```

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 %>% dplyr::select(country, lifeExp, year) %>% dplyr::filter(gm$lifeExp >= 80) %>% dplyr::group_by(yea
```

```
## # A tibble: 22 x 3
## # Groups:   year, country [22]
##    country          lifeExp  year
```

```
##    <chr>                <dbl> <int>
##  1 Australia             80.4  2002
##  2 Australia             81.2  2007
##  3 Canada                80.7  2007
##  4 France                80.7  2007
##  5 Hong Kong, China      80    1997
##  6 Hong Kong, China      81.5  2002
##  7 Hong Kong, China      82.2  2007
##  8 Iceland               80.5  2002
##  9 Iceland               81.8  2007
## 10 Israel                80.7  2007
## # ... with 12 more rows
```

Question 8

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.

```r
#install.packages("hflights")
# Load the library
library(hflights)
View(hflights)
```

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

```r
hf_dataframe <- hflights
head(hflights, 20)
```

```
##      Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum
## 5424 2011     1          1         6    1400    1500            AA       428
## 5425 2011     1          2         7    1401    1501            AA       428
## 5426 2011     1          3         1    1352    1502            AA       428
## 5427 2011     1          4         2    1403    1513            AA       428
## 5428 2011     1          5         3    1405    1507            AA       428
## 5429 2011     1          6         4    1359    1503            AA       428
## 5430 2011     1          7         5    1359    1509            AA       428
## 5431 2011     1          8         6    1355    1454            AA       428
## 5432 2011     1          9         7    1443    1554            AA       428
## 5433 2011     1         10         1    1443    1553            AA       428
## 5434 2011     1         11         2    1429    1539            AA       428
## 5435 2011     1         12         3    1419    1515            AA       428
## 5436 2011     1         13         4    1358    1501            AA       428
## 5437 2011     1         14         5    1357    1504            AA       428
## 5438 2011     1         15         6    1359    1459            AA       428
## 5439 2011     1         16         7    1359    1509            AA       428
## 5440 2011     1         17         1    1530    1634            AA       428
## 5441 2011     1         18         2    1408    1508            AA       428
## 5442 2011     1         19         3    1356    1503            AA       428
## 5443 2011     1         20         4    1507    1622            AA       428
##      TailNum ActualElapsedTime AirTime ArrDelay DepDelay Origin Dest Distance
## 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, [we are displaying only 50 rows as datasize is huge]

```
hflights %>% filter(hflights$Month == 1, hflights$DayofMonth == 1) %>% head(50)
```

```
##    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
## 6  2011     1          1         6    1211    1325            AA      1820
## 7  2011     1          1         6     557     906            AA      1994
## 8  2011     1          1         6    1824    2106            AS       731
## 9  2011     1          1         6     654    1124            B6       620
```

```
## 10 2011     1          1         6    1639    2110            B6       622
## 11 2011     1          1         6     942    1356            CO         1
## 12 2011     1          1         6    1845    1947            CO         5
## 13 2011     1          1         6    1533    1634            CO         6
## 14 2011     1          1         6    1459    1602            CO        33
## 15 2011     1          1         6    1551    1650            CO        35
## 16 2011     1          1         6    1923    2115            CO        47
## 17 2011     1          1         6     748     946            CO        52
## 18 2011     1          1         6    1807    1939            CO        59
## 19 2011     1          1         6    1218    1623            CO        60
## 20 2011     1          1         6    1446    1906            CO        62
## 21 2011     1          1         6    1145    1612            CO        73
## 22 2011     1          1         6    1447    1925            CO        77
## 23 2011     1          1         6     558    1006            CO        89
## 24 2011     1          1         6    1049    1458            CO       106
## 25 2011     1          1         6    1428    1608            CO       137
## 26 2011     1          1         6    1322    1552            CO       146
## 27 2011     1          1         6    1933    2106            CO       150
## 28 2011     1          1         6     935    1214            CO       170
## 29 2011     1          1         6    1927    2256            CO       190
## 30 2011     1          1         6     850    1025            CO       195
## 31 2011     1          1         6    1234    1358            CO       197
## 32 2011     1          1         6     721     851            CO       199
## 33 2011     1          1         6    1244    1547            CO       206
## 34 2011     1          1         6    2133       4            CO       209
## 35 2011     1          1         6    1554    2019            CO       210
## 36 2011     1          1         6    1038    1646            CO       212
## 37 2011     1          1         6     902    1054            CO       220
## 38 2011     1          1         6    1017    1419            CO       226
## 39 2011     1          1         6    1550    1917            CO       244
## 40 2011     1          1         6    1607    1838            CO       246
## 41 2011     1          1         6    1917    2248            CO       250
## 42 2011     1          1         6    1136    1357            CO       252
## 43 2011     1          1         6    1226    1459            CO       267
## 44 2011     1          1         6    1205    1420            CO       270
## 45 2011     1          1         6     717    1145            CO       282
## 46 2011     1          1         6    1742    2108            CO       286
## 47 2011     1          1         6    1919    2035            CO       297
## 48 2011     1          1         6    2058    2226            CO       299
## 49 2011     1          1         6     858    1136            CO       309
## 50 2011     1          1         6    1944    2349            CO       310
##    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
## 6   N593AA                74      39       15        6    IAH  DFW      224
## 7   N3BBAA               129     113       -9       -3    IAH  MIA      964
## 8   N614AS               282     255       -4       -1    IAH  SEA     1874
## 9   N324JB               210     181        5       -6    HOU  JFK     1428
## 10  N324JB               211     188       61       54    HOU  JFK     1428
## 11  N69063               494     466        6       17    IAH  HNL     3904
## 12  N29717                62      43       13       15    IAH  MSY      305
```

```
## 13  N47414              61     36     14      8   IAH  SAT    191
## 14  N62631              63     41     19     18   IAH  MSY    305
## 15  N14653              59     30     23     16   IAH  AUS    140
## 16  N74856             232    191     15     13   IAH  LAX   1379
## 17  N19130             238    205     20      3   IAH  LAX   1379
## 18  N35204             152    130     27     27   IAH  DEN    862
## 19  N67157             185    162     -2     13   IAH  EWR   1400
## 20  N26123             200    163     17     21   IAH  EWR   1400
## 21  N76065             507    486      2      0   IAH  HNL   3904
## 22  N76062             518    488     35     22   IAH  HNL   3904
## 23  N73406             188    161     -6     -2   IAH  EWR   1400
## 24  N68159             189    162     -9      4   IAH  EWR   1400
## 25  N39415             220    200     -2      3   IAH  LAX   1379
## 26  N72405             150    117      2      7   IAH  ORD    925
## 27  N78506             213    186     49     51   IAH  ONT   1334
## 28  N56859             279    251     59     35   IAH  SFO   1635
## 29  N33209             149    118     33     22   IAH  MIA    964
## 30  N24729             215    195      5     15   IAH  LAX   1379
## 31  N73259             204    175     12     10   IAH  LAS   1222
## 32  N73278             150    118     -6      1   IAH  DEN    862
## 33  N75429             123     98     -1     -1   IAH  TPA    787
## 34  N37422             271    247      3      3   IAH  PDX   1825
## 35  N75436             205    177      4      4   IAH  EWR   1400
## 36  N75426             248    223     24     23   IAH  SJU   2007
## 37  N19136             172    141     -2      2   IAH  PHX   1009
## 38  N17619             182    148      4     -4   IAH  BWI   1235
## 39  N76265             147    118     -6      5   IAH  CLE   1091
## 40  N37427             151    112     16     22   IAH  ORD    925
## 41  N14613             151    125      3      7   IAH  RDU   1043
## 42  N14613             201    177      9      4   IAH  SLC   1195
## 43  N12216             273    253     -6      0   IAH  SEA   1874
## 44  N16217             255    234     40     40   IAH  SFO   1635
## 45  N14639             208    176    -23     -8   IAH  BOS   1597
## 46  N32626             146    109     41     27   IAH  MCO    853
## 47  N35407             196    175     -6      4   IAH  LAS   1222
## 48  N37293             148    128      4      3   IAH  DEN    862
## 49  N35407             278    250      8      8   IAH  PDX   1825
## 50  N67058             185    156     28     44   IAH  EWR   1400
##    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
## 6       6      29         0                         0
## 7       5      11         0                         0
## 8       7      20         0                         0
## 9       6      23         0                         0
## 10     12      11         0                         0
## 11      5      23         0                         0
## 12      3      16         0                         0
## 13      5      20         0                         0
## 14      2      20         0                         0
## 15      5      24         0                         0
```

```
## 16      13       28          0                          0
## 17      16       17          0                          0
## 18       7       15          0                          0
## 19       8       15          0                          0
## 20       9       28          0                          0
## 21       4       17          0                          0
## 22       8       22          0                          0
## 23       7       20          0                          0
## 24      11       16          0                          0
## 25       5       15          0                          0
## 26       9       24          0                          0
## 27       5       22          0                          0
## 28       6       22          0                          0
## 29       4       27          0                          0
## 30       6       14          0                          0
## 31      10       19          0                          0
## 32      12       20          0                          0
## 33       6       19          0                          0
## 34       6       18          0                          0
## 35       8       20          0                          0
## 36       6       19          0                          0
## 37       5       26          0                          0
## 38       6       28          0                          0
## 39       4       25          0                          0
## 40      16       23          0                          0
## 41       7       19          0                          0
## 42       6       18          0                          0
## 43       4       16          0                          0
## 44       4       17          0                          0
## 45       7       25          0                          0
## 46       8       29          0                          0
## 47       8       13          0                          0
## 48       6       14          0                          0
## 49       5       23          0                          0
## 50       9       20          0                          0
```

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

```
hflights %>% filter(UniqueCarrier == "UA" | UniqueCarrier == "AA") %>% head(50)
```

```
##    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
## 6  2011     1          6         4    1359    1503            AA       428
## 7  2011     1          7         5    1359    1509            AA       428
## 8  2011     1          8         6    1355    1454            AA       428
## 9  2011     1          9         7    1443    1554            AA       428
## 10 2011     1         10         1    1443    1553            AA       428
## 11 2011     1         11         2    1429    1539            AA       428
## 12 2011     1         12         3    1419    1515            AA       428
```

```
## 13 2011     1          13          4    1358    1501            AA       428
## 14 2011     1          14          5    1357    1504            AA       428
## 15 2011     1          15          6    1359    1459            AA       428
## 16 2011     1          16          7    1359    1509            AA       428
## 17 2011     1          17          1    1530    1634            AA       428
## 18 2011     1          18          2    1408    1508            AA       428
## 19 2011     1          19          3    1356    1503            AA       428
## 20 2011     1          20          4    1507    1622            AA       428
## 21 2011     1          21          5    1357    1459            AA       428
## 22 2011     1          22          6    1355    1456            AA       428
## 23 2011     1          23          7    1356    1501            AA       428
## 24 2011     1          24          1    1356    1513            AA       428
## 25 2011     1          25          2    1352    1452            AA       428
## 26 2011     1          26          3    1353    1455            AA       428
## 27 2011     1          27          4    1356    1458            AA       428
## 28 2011     1          28          5    1359    1505            AA       428
## 29 2011     1          29          6    1355    1455            AA       428
## 30 2011     1          30          7    1359    1456            AA       428
## 31 2011     1          31          1    1441    1553            AA       428
## 32 2011     1           1          6     728     840            AA       460
## 33 2011     1           2          7     719     821            AA       460
## 34 2011     1           3          1     717     834            AA       460
## 35 2011     1           4          2     714     821            AA       460
## 36 2011     1           5          3     718     822            AA       460
## 37 2011     1           6          4     719     821            AA       460
## 38 2011     1           7          5     711     827            AA       460
## 39 2011     1           8          6     713     805            AA       460
## 40 2011     1           9          7     714     829            AA       460
## 41 2011     1          10          1     715     818            AA       460
## 42 2011     1          11          2     717     820            AA       460
## 43 2011     1          12          3     714     814            AA       460
## 44 2011     1          13          4     722     841            AA       460
## 45 2011     1          14          5     715     828            AA       460
## 46 2011     1          15          6     719     833            AA       460
## 47 2011     1          16          7     743     843            AA       460
## 48 2011     1          17          1     724     842            AA       460
## 49 2011     1          18          2     721     827            AA       460
## 50 2011     1          19          3     714     833            AA       460
##    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
## 6   N262AA                64      45       -7       -1    IAH  DFW      224
## 7   N493AA                70      43       -1       -1    IAH  DFW      224
## 8   N477AA                59      40      -16       -5    IAH  DFW      224
## 9   N476AA                71      41       44       43    IAH  DFW      224
## 10  N504AA                70      45       43       43    IAH  DFW      224
## 11  N565AA                70      42       29       29    IAH  DFW      224
## 12  N577AA                56      41        5       19    IAH  DFW      224
## 13  N476AA                63      44       -9       -2    IAH  DFW      224
## 14  N552AA                67      47       -6       -3    IAH  DFW      224
## 15  N462AA                60      44      -11       -1    IAH  DFW      224
```

```
## 16  N555AA                70       41        -1        -1    IAH  DFW      224
## 17  N518AA                64       48        84        90    IAH  DFW      224
## 18  N507AA                60       42        -2         8    IAH  DFW      224
## 19  N523AA                67       46        -7        -4    IAH  DFW      224
## 20  N425AA                75       42        72        67    IAH  DFW      224
## 21  N251AA                62       47       -11        -3    IAH  DFW      224
## 22  N551AA                61       44       -14        -5    IAH  DFW      224
## 23  N479AA                65       40        -9        -4    IAH  DFW      224
## 24  N531AA                77       43         3        -4    IAH  DFW      224
## 25  N561AA                60       40       -18        -8    IAH  DFW      224
## 26  N541AA                62       40       -15        -7    IAH  DFW      224
## 27  N512AA                62       40       -12        -4    IAH  DFW      224
## 28  N4UBAA                66       46        -5        -1    IAH  DFW      224
## 29  N491AA                60       46       -15        -5    IAH  DFW      224
## 30  N561AA                57       39       -14        -1    IAH  DFW      224
## 31  N505AA                72       39        43        41    IAH  DFW      224
## 32  N520AA                72       41         5         8    IAH  DFW      224
## 33  N537AA                62       43       -14        -1    IAH  DFW      224
## 34  N512AA                77       46        -1        -3    IAH  DFW      224
## 35  N478AA                67       46       -14        -6    IAH  DFW      224
## 36  N551AA                64       44       -13        -2    IAH  DFW      224
## 37  N251AA                62       44       -14        -1    IAH  DFW      224
## 38  N478AA                76       42        -8        -9    IAH  DFW      224
## 39  N550AA                52       40       -30        -7    IAH  DFW      224
## 40  N586AA                75       51        -6        -6    IAH  DFW      224
## 41  N587AA                63       44       -17        -5    IAH  DFW      224
## 42  N574AA                63       44       -15        -3    IAH  DFW      224
## 43  N580AA                60       42       -21        -6    IAH  DFW      224
## 44  N586AA                79       49         6         2    IAH  DFW      224
## 45  N468AA                73       47        -7        -5    IAH  DFW      224
## 46  N251AA                74       49        -2        -1    IAH  DFW      224
## 47  N546AA                60       45         8        23    IAH  DFW      224
## 48  N559AA                78       54         7         4    IAH  DFW      224
## 49  N558AA                66       46        -8         1    IAH  DFW      224
## 50  N574AA                79       51        -2        -6    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
## 6       6      13         0                         0
## 7      12      15         0                         0
## 8       7      12         0                         0
## 9       8      22         0                         0
## 10      6      19         0                         0
## 11      8      20         0                         0
## 12      4      11         0                         0
## 13      6      13         0                         0
## 14      5      15         0                         0
## 15      6      10         0                         0
## 16     12      17         0                         0
## 17      8       8         0                         0
## 18      7      11         0                         0
```

```
## 19      10      11           0                              0
## 20       9      24           0                              0
## 21       6       9           0                              0
## 22       9       8           0                              0
## 23       7      18           0                              0
## 24       6      28           0                              0
## 25       7      13           0                              0
## 26       8      14           0                              0
## 27      12      10           0                              0
## 28       8      12           0                              0
## 29       7       7           0                              0
## 30       7      11           0                              0
## 31       8      25           0                              0
## 32       6      25           0                              0
## 33       9      10           0                              0
## 34      21      10           0                              0
## 35       6      15           0                              0
## 36       7      13           0                              0
## 37       8      10           0                              0
## 38      24      10           0                              0
## 39       3       9           0                              0
## 40      11      13           0                              0
## 41       8      11           0                              0
## 42       7      12           0                              0
## 43      10       8           0                              0
## 44      16      14           0                              0
## 45      15      11           0                              0
## 46      12      13           0                              0
## 47       5      10           0                              0
## 48      12      12           0                              0
## 49       7      13           0                              0
## 50      14      14           0                              0
```

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".

```r
x <- hflights %>% select(Year, Month, DayofMonth, contains("Taxi"), contains("Delay")) %>% head(50)

x
```

```
##      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
## 5429 2011     1          6      6      13       -7       -1
## 5430 2011     1          7     12      15       -1       -1
## 5431 2011     1          8      7      12      -16       -5
## 5432 2011     1          9      8      22       44       43
## 5433 2011     1         10      6      19       43       43
## 5434 2011     1         11      8      20       29       29
## 5435 2011     1         12      4      11        5       19
## 5436 2011     1         13      6      13       -9       -2
```

```
## 5437 2011      1          14      5      15       -6       -3
## 5438 2011      1          15      6      10      -11       -1
## 5439 2011      1          16     12      17       -1       -1
## 5440 2011      1          17      8       8       84       90
## 5441 2011      1          18      7      11       -2        8
## 5442 2011      1          19     10      11       -7       -4
## 5443 2011      1          20      9      24       72       67
## 5444 2011      1          21      6       9      -11       -3
## 5445 2011      1          22      9       8      -14       -5
## 5446 2011      1          23      7      18       -9       -4
## 5447 2011      1          24      6      28        3       -4
## 5448 2011      1          25      7      13      -18       -8
## 5449 2011      1          26      8      14      -15       -7
## 5450 2011      1          27     12      10      -12       -4
## 5451 2011      1          28      8      12       -5       -1
## 5452 2011      1          29      7       7      -15       -5
## 5453 2011      1          30      7      11      -14       -1
## 5454 2011      1          31      8      25       43       41
## 6343 2011      1           1      6      25        5        8
## 6344 2011      1           2      9      10      -14       -1
## 6345 2011      1           3     21      10       -1       -3
## 6346 2011      1           4      6      15      -14       -6
## 6347 2011      1           5      7      13      -13       -2
## 6348 2011      1           6      8      10      -14       -1
## 6349 2011      1           7     24      10       -8       -9
## 6350 2011      1           8      3       9      -30       -7
## 6351 2011      1           9     11      13       -6       -6
## 6352 2011      1          10      8      11      -17       -5
## 6353 2011      1          11      7      12      -15       -3
## 6354 2011      1          12     10       8      -21       -6
## 6355 2011      1          13     16      14        6        2
## 6356 2011      1          14     15      11       -7       -5
## 6357 2011      1          15     12      13       -2       -1
## 6358 2011      1          16      5      10        8       23
## 6359 2011      1          17     12      12        7        4
## 6360 2011      1          18      7      13       -8        1
## 6361 2011      1          19     14      14       -2       -6
```

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

```
y <- hflights %>% select(contains("Depature Time"),contains("Arrival Time"), contains("Flight Number"))


y
```

```
## data frame with 0 columns and 50 rows
```

8 (f) Print all the aircrafts carriers whose departure time is delayed more than 60 minutes

```
hflights %>% filter(hflights$DepDelay > 60) %>% group_by(UniqueCarrier) %>% distinct(UniqueCarrier) %>%
```

```
## # A tibble: 14 x 1
```

```
## # Groups:   UniqueCarrier [14]
##    UniqueCarrier
##    <chr>
##  1 AA
##  2 AS
##  3 B6
##  4 CO
##  5 DL
##  6 OO
##  7 UA
##  8 US
##  9 WN
## 10 EV
## 11 F9
## 12 FL
## 13 MQ
## 14 XE
```

8 (g) Look at the carriers with their departures delays and sort them based on their departure delays

```
library(dplyr)

hflights %>% select(UniqueCarrier,DepDelay) %>% arrange(desc(DepDelay)) %>% head(50)
```

```
##    UniqueCarrier DepDelay
## 1             CO      981
## 2             AA      970
## 3             MQ      931
## 4             UA      869
## 5             MQ      814
## 6             MQ      803
## 7             CO      780
## 8             CO      758
## 9             DL      730
## 10            MQ      691
## 11            AA      677
## 12            AA      653
## 13            XE      628
## 14            UA      588
## 15            CO      576
## 16            UA      563
## 17            WN      548
## 18            UA      535
## 19            AA      525
## 20            MQ      520
## 21            XE      511
## 22            FL      507
## 23            WN      503
## 24            WN      499
## 25            DL      497
## 26            FL      493
## 27            UA      490
## 28            DL      488
```

```
## 29              CO      488
## 30              UA      487
## 31              EV      479
## 32              WN      476
## 33              CO      472
## 34              EV      465
## 35              DL      460
## 36              DL      458
## 37              MQ      440
## 38              MQ      427
## 39              US      425
## 40              CO      420
## 41              WN      419
## 42              XE      417
## 43              XE      416
## 44              XE      406
## 45              XE      400
## 46              XE      398
## 47              XE      394
## 48              XE      391
## 49              WN      389
## 50              UA      387
```

IDS572 Business Data Mining

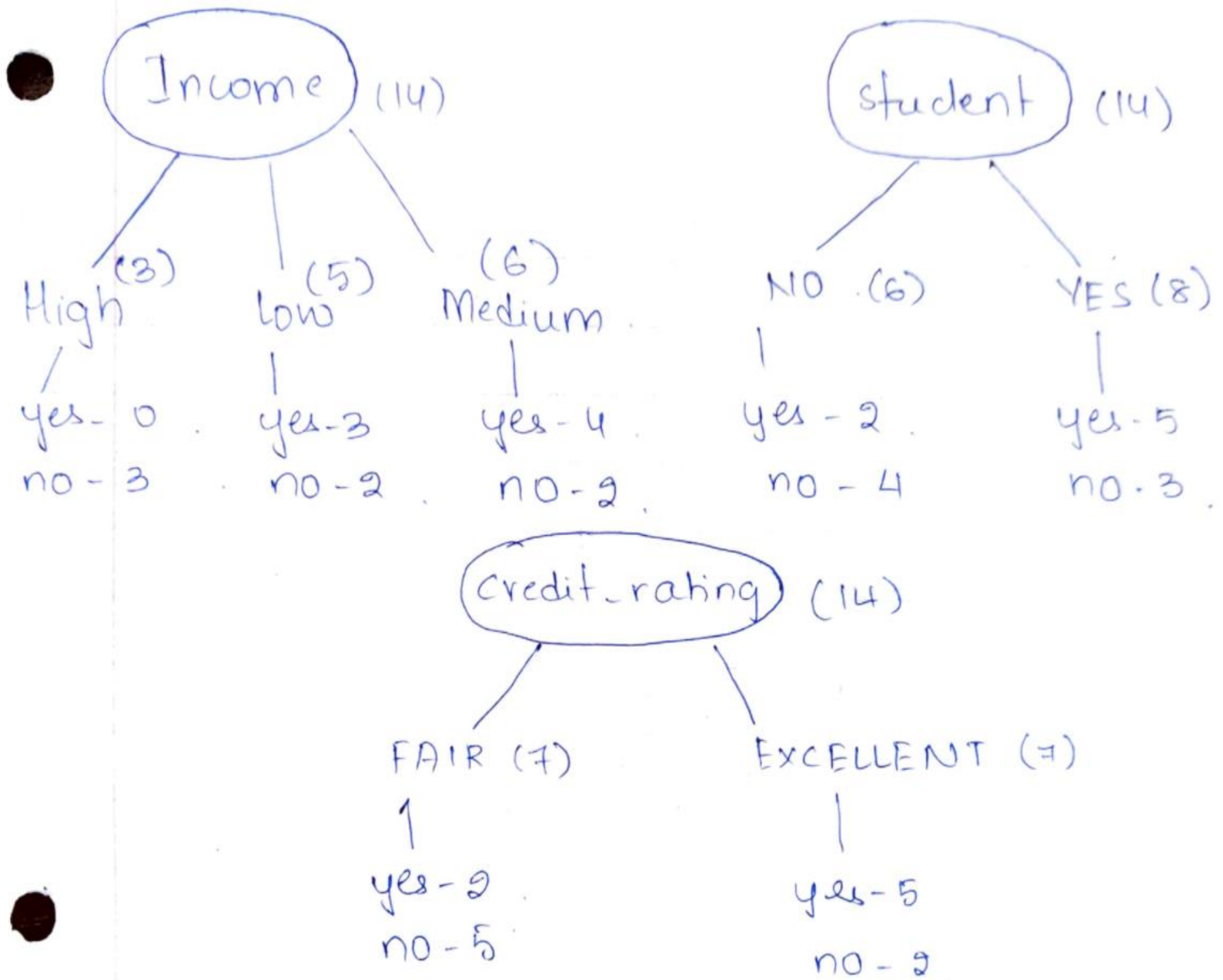
Team members:

- **Aditya Madhavi – 660225279**
- **Sayali Bonawale - 656488690**
- **Navya Yadagiri - 6747883845**

Q 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?



a. Decision rules:

Attribute	Rule	Error rate	Total Error Rate
INCOME	INCOME = HIGH-> BUYS_COMP = NO	0/3	4/14
	INCOME = LOW -> BUYS_COMP = YES	2/5	
	INCOME = MED -> BUYS_COMP = YES	2/6	
STUDENT	STUDENT = NO -> BUYS_COMP = NO	2/6	5/14
	STUDENT = YES -> BUYS_COMP = YES	3/8	
CREDIT_RATING	CREDIT_RATING = FAIR -> BUYS_COMP = NO	2/7	4/14
	CREDIT_RATING = EXCELLENT -> BUYS_COMP = YES	2/7	

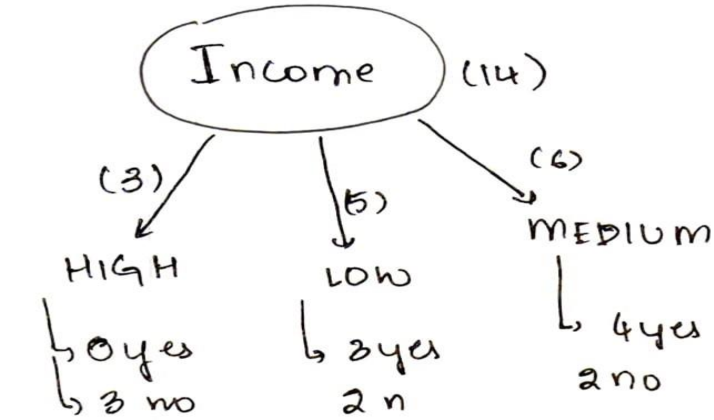
So, from the above, we conclude that the INCOME and CREDIT_RATING attribute decision rule has the lowest misclassification rate of 4/14, and among the three-attribute set of decision rules =, they have the lowest misclassification rate/ error rate

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?

Target Variable – BUYS_COMPUTER

Computing Gini Index for Income Attribute

(contd)



Computing Gini Index :

$$\boxed{\text{Gini Index} = 1 - \sum_{i=1}^{n} (P_i)^2}$$ P_i = Probability of ith value .

Gini Index for Income Branch - High .

$$1 - \left(\frac{0}{3}\right)^2 - \left(\frac{3}{3}\right)^2 = 1 - 0 - 1 = 0$$

Probability of yes $= \left(\frac{0}{3}\right)$ for Income = High

Similarly, $1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2 \Rightarrow 1 - \frac{9}{25} - \frac{4}{25}$

$\Rightarrow \frac{25 - 13}{25} = \frac{12}{25} = 0.48$ for Income = LOW

Gini Index for Income = Medium

$$1 - \left(\frac{2}{6}\right)^2 - \left(\frac{4}{6}\right)^2 \Rightarrow 1 - \frac{4}{36} - \frac{16}{36}$$

$$\Rightarrow \frac{36 - 20}{36} = \frac{16}{36} = 0.44$$

Weighted Sum of Gin Indices :

Probability of Income = HIGH $= \frac{3}{14}$

Probability of Income = LOW $= \frac{5}{14}$

Probability of Income = MEDIUM $= \frac{6}{14}$

Weighted Sum of Gini Indices (INCOME) $= \frac{3}{14}(0) + \frac{5}{14}(0.48) + \frac{6}{14}(0.44)$

$= 0.36$

Computing the Gini Index for Student Attribute:

Computing Gini Index for Student Attribute

student (14)

NO (6) — 2 yes, 4 No.

YES (8) — 5 yes, 3 No.

Target

Gini Index for:

i) Branch : student = NO

$$1 - \left(\frac{2}{6}\right)^2 - \left(\frac{4}{6}\right)^2$$

$$= 1 - \left(\frac{4}{36}\right) - \frac{16}{36}$$

$$1 - \frac{20}{36} \Rightarrow \frac{16}{36} = 0.44$$

ii) Gini Index for student = yes.

$$1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2$$

$$\Rightarrow 1 - \frac{25}{64} - \frac{9}{64} \Rightarrow 1 - \frac{34}{64} = \frac{30}{64} = 0.46875$$

Probability of student = yes = $\frac{8}{14}$

Probability of student = NO = $\frac{6}{14}$

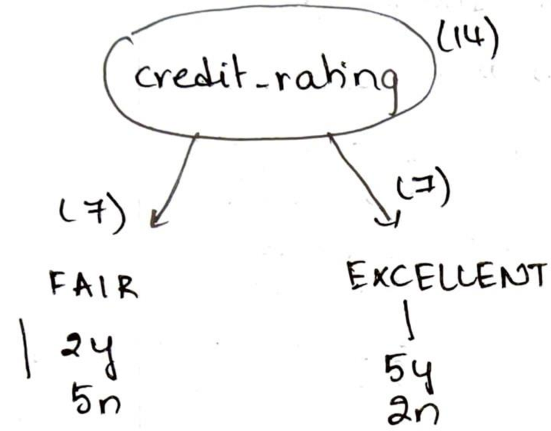
Gini Index (Student)

$$= \frac{6}{14}(0.44) + \frac{8}{14}(0.4687)$$

$$= 0.456$$

Computing the Gini Index for **CREDIT_RATING** Attribute:

Computing Gini Index for Credit Rating Attribute



Gini Index for

(1) credit_rating = FAIR

$$1 - \left(\frac{2}{7}\right)^2 - \left(\frac{5}{7}\right)^2$$

$$\Rightarrow 1 - \frac{4}{49} - \frac{25}{49}$$

$$\Rightarrow \frac{49-29}{49} = \frac{20}{49}$$

$$= 0.408$$

Similarly ; credit_rating = EXCELLENT

$$1 - \left(\frac{5}{7}\right)^2 - \left(\frac{2}{7}\right)^2$$

$$\Rightarrow 1 - \frac{25}{49} - \frac{4}{49} = 0.408$$

Gini Index (credit_rating)

$$= 0.408 \left(\frac{7}{14}\right) + \frac{7}{14}(0.408)$$

Since,

Probability of credit_rating = "Fair" = 7/14

Probability of credit_rating = "excellent" = 7/14

So, after computing the Gini index for all the attribute, we select the one with least Gini index as the root node

- Gini (Income) = 0.36
- Gini (Student) = 0.456
- Gini (Credit_rating) = 0.408

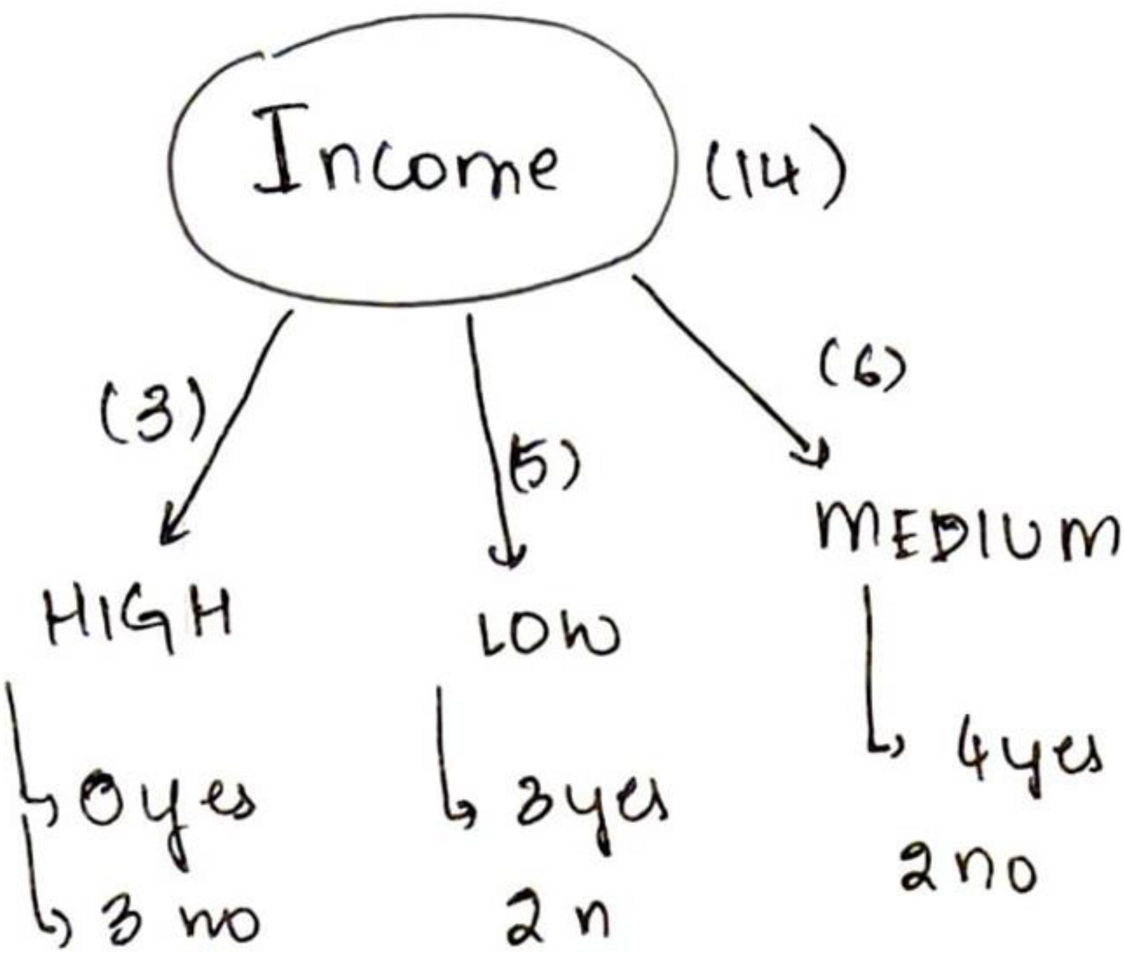
Since, Income has the least Gini index for Income attribute is small, compared to the other two attributes, we choose the Income as the root node.

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

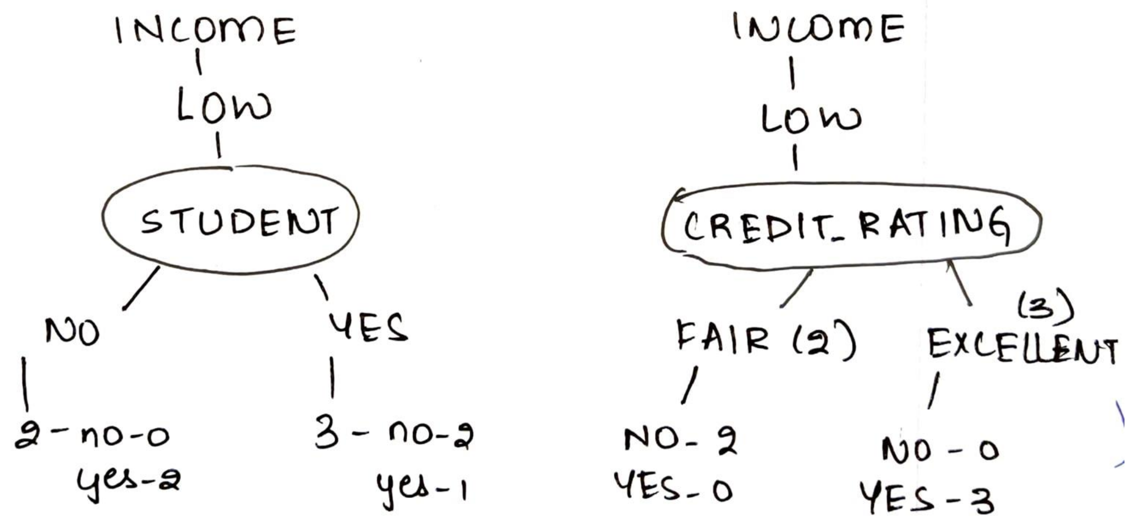
From the above, we conclude that the Root node is Income since it has the least Gini index of 0.36.

Now, to decide on the second level of the decision tree, we

Since the subtree of Income attribute, has one of its branches leading to a pure subset, with INCOME = HIGH, BUYS_COMPUTER = NO, we choose the second level of the decision tree either with branch LOW or MEDIUM.



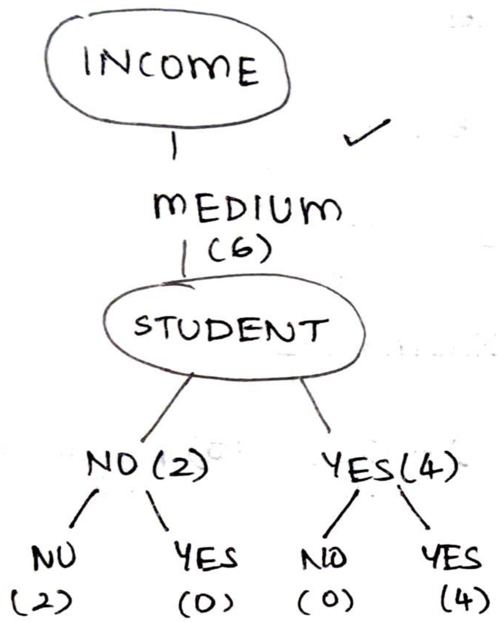
lets consider the following scenarios,
If INCOME = LOW, we choose either student or credit_rating, depending on which gives a purer subset.
If, INCOME = LOW

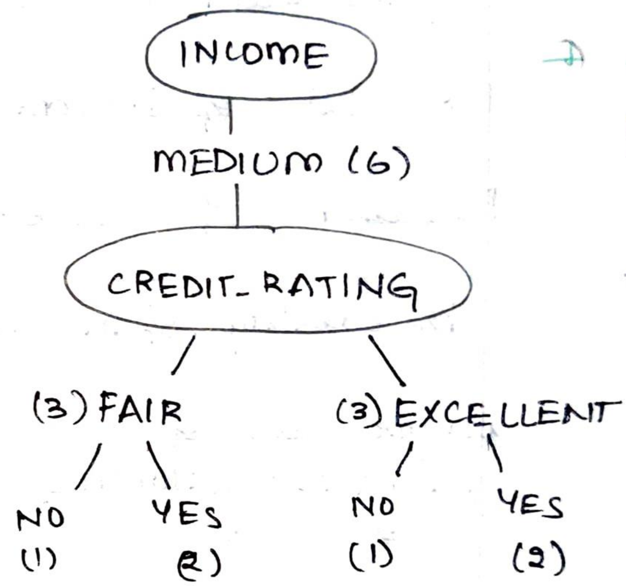


As, shown above if the INCOME = LOW, and Student Attribute as the second level of the decision tree, we cannot achieve a purer subset, but when we have the INCOME = LOW and consider the CREDIT_RATING as the second level of decision tree, we can achieve purer subset. As follows,

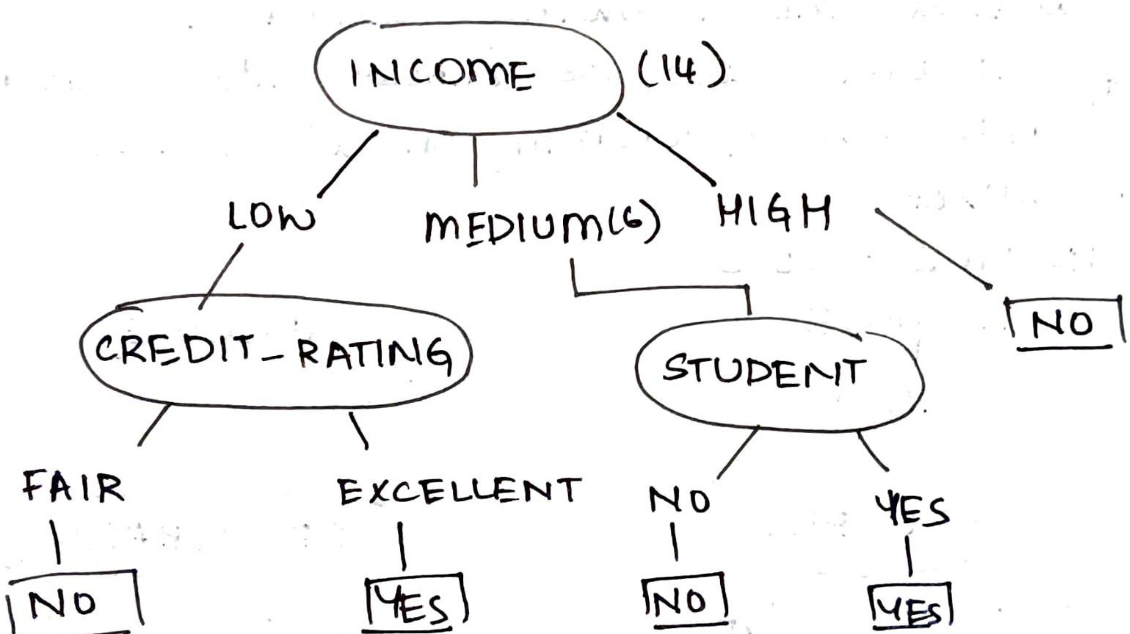
- INCOME= LOW & CREDIT_RATING = FAIR -> BUYS_COMPUTER = NO
- INCOME = LOW & CREDIT_RATING = EXCELLENT-> BUYS_COMPUTER = YES

Similarly, if we consider INCOME = MEDIUM, and if we consider the CREDIT_RATING as the second level of the decision tree, we would not be able to achieve a purer subset, but instead if we when we have INCOME = MEDIUM and STUDENT attribute as the second level of the decision tree as shown be





Final Decision tree:



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

- INCOME = MEDIUM & STUDENT = YES -> BUYS_COMPUTER = YES
- INCOME = MEDIUM & STUDENT = NO -> BUYS_COMPUTER = NO

Since we start with the root node – Income and end with a student node having terminal nodes which are clear or purer subset, we can justify them to be strong student decision rules.

First rule: Support = 4/14 AND confidence 4/4 = 100%

Second Rule: Support = 2/ 14, Confidence = 2/2 = 100%

Since we are getting confidence as 100% for both the rules, we can conclude that these two student decision rules are strong and have 100% confidence.

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

	INCOME	STUDENT	CREDIT_RATING	BUYS_COMPUTER	BUYS_COMPUTER USING DECISION TREE
1	High	No	Fair	No	No
2	High	No	Excellent	No	No
3	Low	No	Excellent	Yes	Yes
4	Medium	No	Fair	No	No
5	Low	Yes	Fair	No	No
6	Low	Yes	Excellent	Yes	Yes
7	Low	No	Excellent	Yes	Yes
8	Medium	Yes	Fair	Yes	Yes
9	Low	Yes	Fair	No	No
10	Medium	Yes	Fair	Yes	Yes
11	Medium	Yes	Excellent	Yes	Yes
12	Medium	No	Excellent	No	No
13	High	Yes	Fair	No	No
14	Medium	Yes	Excellent	Yes	Yes

Accuracy is 100%, since all the training data set instances match the decision rules of the decision tree constructed.