291_Homework-2

```
setwd("D:\\Stat\\R\\Stat291\\Homework_2") #Set directory
```

Question 1-)

```
#1st question
grades <- read.csv("Grades.csv", header = T)

Grade_letter <- vector() #create the empty vector for appending new elements

for(i in 1:length(grades$MT1)){ #iterate through the rows
  av_grade <- .3*grades[i,2]+.3*grades[i,3]+.4*grades[i,4] #calculate the grade
  if(av_grade >= 90 & av_grade <= 100){ #then assign related letter grade
    #to the index i of Grade_letter
    Grade_letter[i] <- "AA"
  }
  else if(av_grade >= 85 & av_grade < 90){
    Grade_letter[i] <- "BA"
  }
  else if(av_grade >= 80 & av_grade < 85){
    Grade_letter[i] <- "BB"
  }
  else if(av_grade >= 75 & av_grade < 80){
    Grade_letter[i] <- "CB"
  }
  else if(av_grade >= 70 & av_grade < 75){
    Grade_letter[i] <- "CC"
  }
  else if(av_grade >= 65 & av_grade < 70){
    Grade_letter[i] <- "DC"
  }
  else if(av_grade >= 60 & av_grade < 65){
    Grade_letter[i] <- "DD"
  }
  else if(av_grade >= 50 & av_grade < 60){
    Grade_letter[i] <- "FD"
  }
  else if(av_grade >= 0 & av_grade < 50){
    Grade_letter[i] <- "FF"
  }
}

df_grades <- data.frame(id = grades$ID, grades = Grade_letter) #2 col dataframe
#that includes ID & Grade_letter
tail(df_grades, 5) #Last 5 rows
```

```
##    id grades
## 16 16     DC
```

```
## 17 17     CB
## 18 18     BB
## 19 19     AA
## 20 20     DC
```

```
table(Grade_letter) #Frequency table
```

```
## Grade_letter
## AA BB CB CC DC DD FF
##  1  2  2  3  5  6  1
```

Question 2-)

```
#2nd question
func <- function(x){
  for(i in 1:x){
    list_temp <- c((1:10)^i)
    thesum <- sum(list_temp)
    print(paste("For p =",i,"Total is",thesum))
  }
}

func(5)
```

```
## [1] "For p = 1 Total is 55"
## [1] "For p = 2 Total is 385"
## [1] "For p = 3 Total is 3025"
## [1] "For p = 4 Total is 25333"
## [1] "For p = 5 Total is 220825"
```

Question 3-)

```
#3rd question
pascal <- function(x){
  if(x == 0){ #if x == 0 we want 0 as return
    return(0)
  }
  row <- c(1) #Assigning this firsthand ables us to return 1 in case x == 1 and
  #not to loose the value in loops :)
  for(i in 1:x){
    print(row)
    row_temp <- row #Temp vector that holds our last row so that we can use
    #it to sum the related parts into new elements
    for(j in 1:length(row)+1){
      if(j==1 | j==length(row)+1){ #the first and last element of a row is 1
        row[j] <- 1
      } else { #Else its the sum of jth and j-1th element of the temp row
        row[j] <- row_temp[j-1]+row_temp[j]
      }
    }
  }
}
pascal(6)
```

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

```
## [1] 1 3 3 1
## [1] 1 4 6 4 1
## [1]  1  5 10 10  5  1
```

```
#I couldnt solve why the shape gets riddled when a higher digit number is introduced
#ex: from 5th to 6th row the middle 2 numbers become 2 digit numbers and the shape becomes weird
#another ex: from 9th to 10th
#Weird.
```

Question 4-)

```
#4th question
prime_check <- function(x){
  if(all.equal(x, as.integer(x)) == TRUE & x > 0){ #if positive integer
    flag = 1
    for(i in 2:floor(sqrt(x))){ #looks for numbers that can divide x (hint is utilized)
      if(x%%i == 0){
        flag = 0
        if(x == 2 | x == 3){ #Special cases
          flag = 1
        }
      }
    }
    if(flag == 1){ #if there was a number that divided x flag became 0 in line 102
      return(TRUE)
    } else if (flag != 1) {
      return(FALSE)
    }
  } else { #if x is not a positive integer it automatically becomes FALSE
    return(FALSE)
  }
}

#non-prime positive integer
prime_check(8)
```

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

```
#prime positive integer
prime_check(7)
```

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

```
#negative integer
prime_check(-4)
```

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

```
#non-integer value
prime_check(pi)
```

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

Question 5-)

```
#5th question

##1-)
agefat <- read.table("agefat.txt", header = T)
```

```r
head(agefat, 5)
```

```
##   Age  Fat Gender
## 1  23 19.2      m
## 2  28 16.6      f
## 3  38 32.5      f
## 4  44 29.1      m
## 5  23 34.5      m
```

```r
##2-)
males <- agefat[agefat$Gender == "m",]
females <- agefat[agefat$Gender == "f",]

males[,3] <- NULL
females[,3] <- NULL

names(males) <- c("age.male", "fat.male")
names(females) <- c("age.female", "fat.female")

##3-)
fivenum(males$age.male)
```

```
## [1] 23 27 34 43 60
```

```r
fivenum(females$age.female)
```

```
## [1] 26 28 34 55 57
```

```r
summary(males$age.male)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.00   27.00   34.00   35.53   43.00   60.00
```

```r
summary(females$age.female)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   26.00   28.00   34.00   38.20   50.75   57.00
```

```r
# These two types of summary creation functions print different values for females
# Dont know wich one is the correct so im putting both of these here.
# The oldest person in class is a male. The youngest person in class is a male.

##4-)

cov(males$age.male, males$fat.male)
```

```
## [1] 19.57238
```

```r
# 19.57238
# This value tells us that There is a relatively big positive relation with
# body fat and age

cov(females$age.female, females$fat.female)
```

```
## [1] 16.76
```

```r
# 16.76
# Same with the male covariance, a positive covariance means older females
# generally has more fat in their bodies.
```

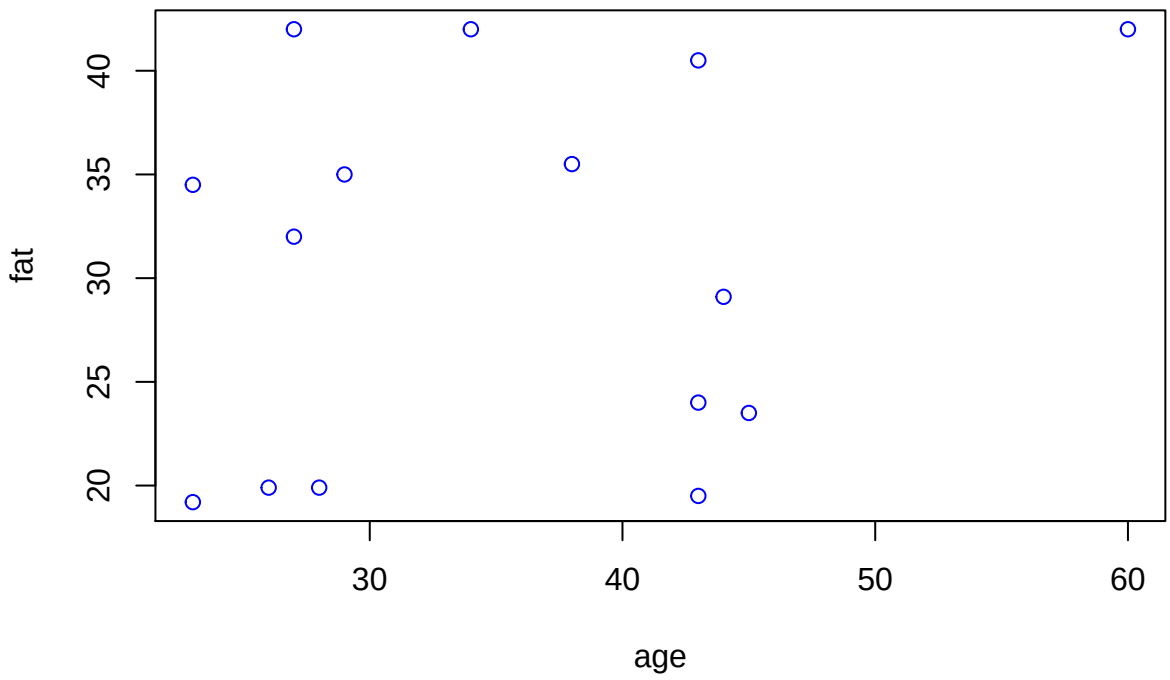
```r
cor(males$age.male, males$fat.male)
```

```
## [1] 0.2049912
```

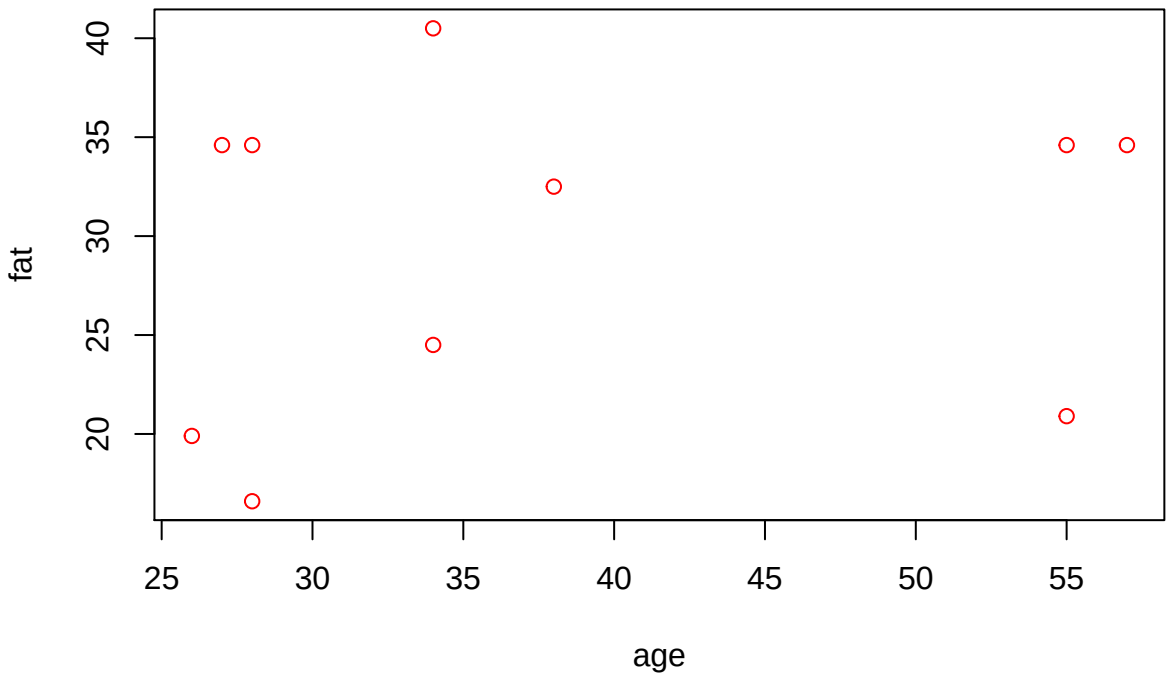
```r
# 0.2049912
# A positive correlation means that the likelihood of variables increasing with
# each other is higher

cor(females$age.female, females$fat.female)
```

```
## [1] 0.1637204
```

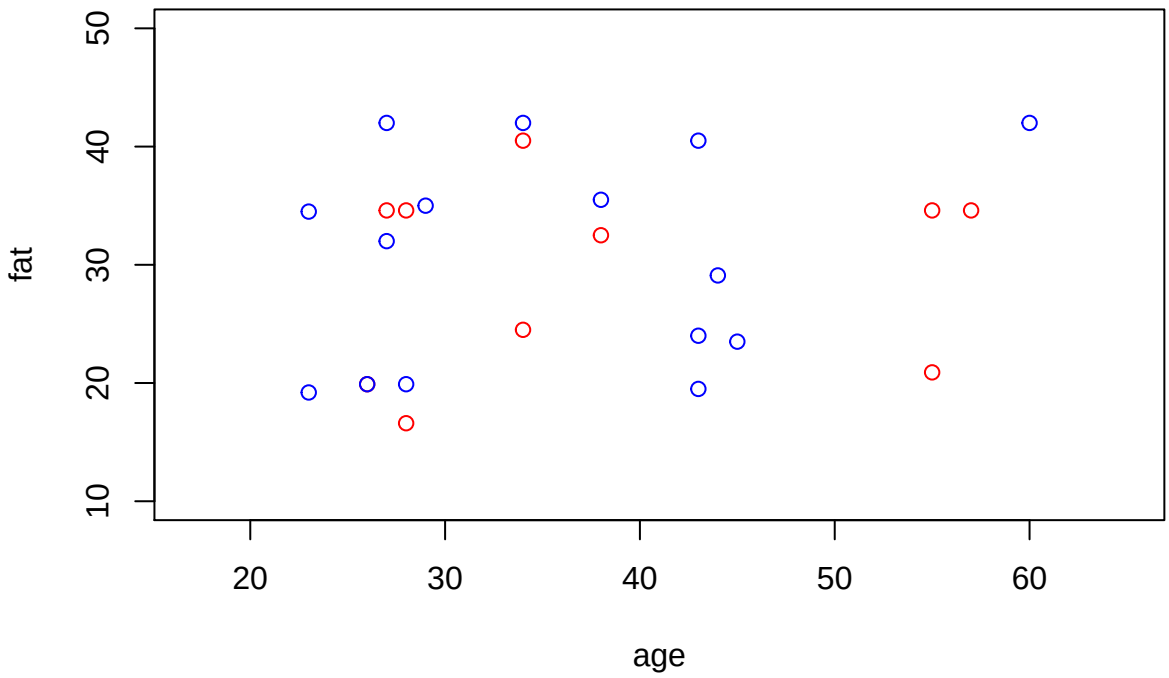
```r
# 0.1637204
# Same with the male correlation, variables generally effect each other positively.

##5-)
#male
plot(males$age.male,males$fat.male, col="blue", xlab = "age", ylab = "fat") #Blue for males
```




```r
#female
plot(females$age.female,females$fat.female, col="red", xlab = "age", ylab = "fat") #Red for females
```




```
##6-)
plot(females$age.female,females$fat.female,col="red", xlab = "age",
     ylab = "fat", xlim = c(17,65), ylim = c(10, 50))
# as the margins are not the same between two plots we set them using xlim, ylim.
# So no value is lost in the plot.
points(males$age.male,males$fat.male,col="blue")
```


fat
10
20
30
40
50
20
30
40
50
60
age