R Assigment 2

Azam Jainullabudin Mohamed & 18201501

14 November 2018

1.Load the lawyers' data into R. What proportion of the lawyers practices litigation law? (Give your answer to 2 decimal places.)

```
Lawyers <- read.csv("Lawyers.csv") #Importing the Lawyer.csv dataset
head(Lawyers)
```

```
##   Seniority Gender  Office Years Age   Practice                   School
## 1   Partner   Male  Boston    31  64 Litigation           Harvard or Yale
## 2   Partner   Male  Boston    32  62  Corporate           Harvard or Yale
## 3   Partner   Male Harvard    13  67 Litigation           Harvard or Yale
## 4   Partner   Male  Boston    31  59  Corporate                    Other
## 5   Partner   Male Harvard    31  59 Litigation University of Connecticut
## 6   Partner   Male Harvard    29  55 Litigation           Harvard or Yale
```

```
Lawyers$Practice
```

```
##  [1] Litigation Corporate  Litigation Corporate  Litigation Litigation
##  [7] Corporate  Litigation Corporate  Corporate  Litigation Corporate
## [13] Litigation Corporate  Corporate  Corporate  Corporate  Litigation
## [19] Corporate  Litigation Litigation Litigation Litigation Litigation
## [25] Corporate  Litigation Litigation Corporate  Corporate  Litigation
## [31] Litigation Litigation Litigation Corporate  Corporate  Litigation
## [37] Corporate  Litigation Litigation Litigation Litigation Corporate
## [43] Litigation Corporate  Corporate  Corporate  Litigation Corporate
## [49] Litigation Corporate  Litigation Litigation Corporate  Litigation
## [55] Litigation Litigation Litigation Litigation Litigation Corporate
## [61] Corporate  Corporate  Corporate  Corporate  Litigation Litigation
## [67] Litigation Litigation Litigation Corporate  Litigation
## Levels: Corporate Litigation
```

```
Lawyers[,"Practice", drop=FALSE] #Extracting the Practice column
```

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

```
## 48  Corporate
## 49 Litigation
## 50  Corporate
## 51 Litigation
## 52 Litigation
## 53  Corporate
## 54 Litigation
## 55 Litigation
## 56 Litigation
## 57 Litigation
## 58 Litigation
## 59 Litigation
## 60  Corporate
## 61  Corporate
## 62  Corporate
## 63  Corporate
## 64  Corporate
## 65 Litigation
## 66 Litigation
## 67 Litigation
## 68 Litigation
## 69 Litigation
## 70  Corporate
## 71 Litigation
```

```
#Proportion of Lawyers practicing ligitation law
ligitation_law <- mean(Lawyers$Practice == "Litigation")
ligitation_law
```

```
## [1] 0.5774648
```

```
#Proportion of Lawyers practicing ligitation law to 2 decimal places
proporation <- round(mean(Lawyers$Practice == "Litigation"), digits = 2)
proporation
```

```
## [1] 0.58
```

2. Is the proportion of lawyers in the Boston office that practice corporate law higher than the proportion of lawyers in the Providence office that practice corporate law?

```
# Lawyers practicing Corporate law at Boston office
Lawyers[,c("Office", "Practice"), drop=FALSE]
```

```
##        Office   Practice
## 1      Boston Litigation
## 2      Boston  Corporate
## 3     Harvard Litigation
## 4      Boston  Corporate
## 5     Harvard Litigation
## 6     Harvard Litigation
## 7     Harvard  Corporate
## 8      Boston Litigation
## 9      Boston  Corporate
## 10     Boston  Corporate
## 11     Boston Litigation
## 12     Boston  Corporate
## 13     Boston Litigation
## 14    Harvard  Corporate
## 15 Providence  Corporate
## 16     Boston  Corporate
## 17     Boston  Corporate
## 18    Harvard Litigation
## 19     Boston  Corporate
## 20     Boston Litigation
## 21     Boston Litigation
## 22     Boston Litigation
## 23     Boston Litigation
## 24     Boston Litigation
## 25    Harvard  Corporate
## 26     Boston Litigation
## 27     Boston Litigation
## 28    Harvard  Corporate
## 29     Boston  Corporate
## 30    Harvard Litigation
## 31    Harvard Litigation
## 32    Harvard Litigation
## 33    Harvard Litigation
## 34     Boston  Corporate
## 35    Harvard  Corporate
## 36     Boston Litigation
## 37 Providence  Corporate
## 38     Boston Litigation
## 39     Boston Litigation
## 40     Boston Litigation
## 41     Boston Litigation
## 42     Boston  Corporate
## 43     Boston Litigation
## 44 Providence  Corporate
## 45     Boston  Corporate
## 46    Harvard  Corporate
## 47 Providence Litigation
```

```
## 48     Boston  Corporate
## 49     Boston Litigation
## 50    Harvard  Corporate
## 51    Harvard Litigation
## 52     Boston Litigation
## 53     Boston  Corporate
## 54     Boston Litigation
## 55     Boston Litigation
## 56     Boston Litigation
## 57     Boston Litigation
## 58    Harvard Litigation
## 59    Harvard Litigation
## 60     Boston  Corporate
## 61     Boston  Corporate
## 62     Boston  Corporate
## 63    Harvard  Corporate
## 64     Boston  Corporate
## 65     Boston Litigation
## 66     Boston Litigation
## 67     Boston Litigation
## 68     Boston Litigation
## 69     Boston Litigation
## 70     Boston  Corporate
## 71     Boston Litigation
```

```
#subset(Lawyers, subset = Practice == 'Litigation')

# Lawyers practicing Corporate law at Boston office
boston_lawyers <- Lawyers[which(Lawyers$Office == 'Boston' & Lawyers$Practice == 'Corp
orate'),]
boston_lawyers
```

```
##    Seniority Gender Office Years Age  Practice                   School
## 2    Partner   Male Boston    32  62 Corporate           Harvard or Yale
## 4    Partner   Male Boston    31  59 Corporate                     Other
## 9    Partner   Male Boston    25  53 Corporate           Harvard or Yale
## 10   Partner   Male Boston    25  53 Corporate                     Other
## 12   Partner   Male Boston    24  52 Corporate University of Connecticut
## 16   Partner   Male Boston    20  46 Corporate           Harvard or Yale
## 17   Partner   Male Boston    23  50 Corporate           Harvard or Yale
## 19   Partner   Male Boston    19  46 Corporate           Harvard or Yale
## 29   Partner Female Boston    10  38 Corporate                     Other
## 34   Partner Female Boston     8  36 Corporate University of Connecticut
## 42 Associate   Male Boston     4  31 Corporate University of Connecticut
## 45 Associate   Male Boston     3  38 Corporate                     Other
## 48 Associate Female Boston     1  35 Corporate                     Other
## 53 Associate   Male Boston    10  38 Corporate                     Other
## 60 Associate Female Boston     2  31 Corporate University of Connecticut
## 61 Associate Female Boston     2  34 Corporate                     Other
## 62 Associate   Male Boston     2  32 Corporate University of Connecticut
## 64 Associate Female Boston     2  45 Corporate University of Connecticut
## 70 Associate   Male Boston     1  31 Corporate University of Connecticut
```

```
# Lawyers practicing Corporate law at Providence office
corporate_lawyers <- Lawyers[which(Lawyers$Office == 'Providence' & Lawyers$Practice =
= 'Corporate'),]
corporate_lawyers
```

```
##    Seniority Gender     Office Years Age  Practice          School
## 15   Partner   Male Providence    21  48 Corporate           Other
## 37 Associate   Male Providence     5  44 Corporate           Other
## 44 Associate Female Providence     5  53 Corporate Harvard or Yale
```

```
#Count of Boston office lawyers
boston_length <- length(boston_lawyers$Office)
boston_length
```

```
## [1] 19
```

```
#Count of Corporate office lawyers
corporate_length <- length(corporate_lawyers$Office)
corporate_length
```

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

```
# Sum of both Boston and Corporate office
total_proporation <- length(boston_lawyers$Office) +length(corporate_lawyers$Office)
total_proporation
```

```
## [1] 22
```

```
#Proporation of Boston office
boston_proportion <- boston_length/total_proporation
boston_proportion
```

```
## [1] 0.8636364
```

```
#Proporation of Corporate office
corporate_length <- corporate_length/total_proporation
corporate_length
```

```
## [1] 0.1363636
```

```
#Verifying if Boston office proporation is greater than Corporate office proportion
boston_proportion > corporate_length
```

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

3.Use the aggregate function to compute the average age of lawyers who practice corporate law and of lawyers who practice litigation law, across the different levels of seniority. Label the columns of the resulting data frame appropriately.

```
# average age of lawyers practising corporate law and litigation law, accross seniorit
y
avg_age <- Lawyers[which(Lawyers$Practice == 'Litigation' | Lawyers$Practice == 'Corpo
rate'),]

# Using Aggregate function to manipulate the mean of the above match
output_mean <- aggregate(avg_age$Age, by=list(avg_age$Practice, avg_age$Seniority), FU
N =mean)

# Renaming the column names of Group1 and Group2
colnames(output_mean)[1] <- "Law Practice"
colnames(output_mean)[2] <- "Seniority level"
output_mean
```

```
##   Law Practice Seniority level        x
## 1    Corporate       Associate 36.71429
## 2   Litigation       Associate 34.61905
## 3    Corporate         Partner 48.50000
## 4   Litigation         Partner 47.70000
```

4. Which office has the youngest median age?

```
# Extracting the lawyers median age
median_age <- median(Lawyers$Age)
median_age
```

```
## [1] 39
```

```
# Office with young median age
young <- Lawyers[which(Lawyers$Age == median_age), "Office"]
young
```

```
## [1] Harvard
## Levels: Boston Harvard Providence
```

Task 2: Writing your own function 1. Write a function which compute the Rosenbrock banana function using a loop. Test the function on the vectors x = (.2, .5) and x = (.2, .5, .1, .6)

```
# Function for manipulating Rosenbrock banana function using loop
rosenbrock <- function(x){
  # Updating the summation end limit as n-1
  n <- length(x)-1
  z <- 0
  s <- 0
  sum <- 0
  for(i in 1:n){
    # Expression for rosenbrock banana function
    s = (100 * (x[i+1] - x[i] * x[i])^2 + (1 - x[i])^2)
    z[i] <- s
  }
  for(j in 1:n){
    # Finding the summation
    sum = sum + z[j]
    print(sum)
  }
}
# Tesing on Vector x
x <- c(0.2,0.5)
print("Rosenbrock function with vector x")
```

```
## [1] "Rosenbrock function with vector x"
```

```
rosenbrock(x)
```

```
## [1] 21.8
```

```
# Tesing on vector y
y <- c(0.2,0.5,0.1,0.6)
print("Rosenbrock function with vector y")
```

```
## [1] "Rosenbrock function with vector y"
```

```
rosenbrock(y)
```

```
## [1] 21.8
## [1] 24.3
## [1] 59.92
```

2. Propose an alternative function that does not use any loop. Test the function on the same two vectors.

```
# Function for manipulating Rosenbrock banana function without using loop
sum <- 0
banana_function <- function(x){
  n <- length(x)-1 # Updating the summation end limit as n-1
  i <- 1:n
  m = 100 * (x[i+1] - x[i] * x[i])^2 + (1 - x[i])^2
  # Cummulative sum for calcating the summation values
  cumsum(m)
}
# Tesing on Vector x
x <- c(0.2,0.5)
print("Rosenbrock function with vector x")
```

```
## [1] "Rosenbrock function with vector x"
```

```
banana_function(x)
```

```
## [1] 21.8
```

```
y <- c(0.2,0.5,0.1,0.6)
print("Rosenbrock function with vector y")
```

```
## [1] "Rosenbrock function with vector y"
```

```
banana_function(y)
```

```
## [1] 21.80 24.30 59.92
```

3.Compare the timings you obtain by repeating the function calls 100 times using the vector x = (.2, .5, .1, .6) as input.

```
# Printing the timings obtained
system.time(replicate(100, rosenbrock(x = c(0.2,0.5,0.1,0.6))))
```

```
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
```

```
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
```

```
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
```

```
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
```

```
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
```

```
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
```

```
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
## [1] 21.8
## [1] 24.3
## [1] 59.92
```

```
##    user  system elapsed
##       0       0       0
```

```
system.time(replicate(100, banana_function(x = c(0.2,0.5,0.1,0.6))))
```

```
##    user  system elapsed
##       0       0       0
```

Task 3: Writing S3 methods 1. Load in the data as an object called DublinAirport. Assign to the DublinAiport object the classes WeatherData and data.frame.

```
#Importing the DublinAirport.csv dataset
DublinAirport <- read.csv("2018_09_Dublin_Airport.csv")
# Assigning the data frame DublinAirport to the class
class(DublinAirport) <- c('WeatherData', 'data.frame')
DublinAirport
```

```
##      ï..date rain maxtp mintp
## 1  01-Sep-18  0.1  20.9  11.8
## 2  02-Sep-18  1.9  23.0  10.4
## 3  03-Sep-18  0.2  16.9   6.6
## 4  04-Sep-18  0.0  17.5   4.6
## 5  05-Sep-18  4.0  19.2   4.9
## 6  06-Sep-18  4.1  15.3   5.8
## 7  07-Sep-18  0.0  15.6   5.4
## 8  08-Sep-18  6.2  17.8  13.0
## 9  09-Sep-18  0.0  17.7  11.9
## 10 10-Sep-18  0.5  16.6  11.5
## 11 11-Sep-18  2.1  16.5  10.6
## 12 12-Sep-18  0.1  16.9   8.4
## 13 13-Sep-18  0.0  16.7   9.2
## 14 14-Sep-18  1.7  16.6  11.1
## 15 15-Sep-18  0.9  16.9  10.9
## 16 16-Sep-18  2.3  18.7  10.1
## 17 17-Sep-18  0.1  20.4  10.0
## 18 18-Sep-18  1.5  21.1  13.6
## 19 19-Sep-18  0.9  17.9  10.3
## 20 20-Sep-18 15.7  12.0   6.0
## 21 21-Sep-18  1.0  14.1   6.1
## 22 22-Sep-18  0.0  11.9   3.4
## 23 23-Sep-18  0.0  13.3   3.9
## 24 24-Sep-18  0.0  13.9   3.5
## 25 25-Sep-18  0.0  15.9   0.4
## 26 26-Sep-18  0.0  19.5  13.4
## 27 27-Sep-18  0.5  16.6   4.2
## 28 28-Sep-18  0.0  13.7   1.5
## 29 29-Sep-18  0.0  14.8   1.9
## 30 30-Sep-18  0.0  12.8   5.1
```

2. Write an S3 summary method for an object of class WeatherData which produces the following statistical summaries for the rain, maxtp, mintp variables: mean, standard deviation, minimum, maximum.

```
# Writing a new summary method for class WeatherData
summary.WeatherData <- function(object){
  # Calculating the mean, sd, max, min value of "Rain" using sapply and
  # printing the result using cat command
  cat('Mean of Rain', sapply(object["rain"], mean), '\n')
  cat('Standart deviation of Rain', sapply(object["rain"], sd), '\n')
  cat('Max of Rain', sapply(object["rain"], max), '\n')
  cat('Min of Rain', sapply(object["rain"], min), '\n')

  # Calculating the mean, sd, max, min value of "maxtp" using sapply and
  # printing the result using cat command
  cat('Mean of maxtp', sapply(object["maxtp"], mean), '\n')
  cat('Standart deviation of Maxtp', sapply(object["maxtp"], sd), '\n')
  cat('Max of Maxtp', sapply(object["maxtp"], max), '\n')
  cat('Min of Maxtp', sapply(object["maxtp"], min), '\n')

  # Calculating the mean, sd, max, min value of "mintp" using sapply and
  # printing the result using cat command
  cat('Mean of Mintp', sapply(object["mintp"], mean), '\n')
  cat('Standart deviation of Mintp', sapply(object["mintp"], sd), '\n')
  cat('Max of Mintp', sapply(object["mintp"], max), '\n')
  cat('Min of Mintp', sapply(object["mintp"], min), '\n')
}
summary.WeatherData(DublinAirport)
```

```
## Mean of Rain 1.46
## Standart deviation of Rain 3.081827
## Max of Rain 15.7
## Min of Rain 0
## Mean of maxtp 16.69
## Standart deviation of Maxtp 2.728313
## Max of Maxtp 23
## Min of Maxtp 11.9
## Mean of Mintp 7.65
## Standart deviation of Mintp 3.851623
## Max of Mintp 13.6
## Min of Mintp 0.4
```

3. Download the new data set 2018 09 Cork Airport.csv from Blackboard, assign the classes WeatherData and data.frame to the object containing the Cork data, and test your function on it. Interpret your findings for Dublin and Cork Airports.

```
#Importing the DublinAirport.csv dataset
CorkAirport <- read.csv("2018_09_Cork_Airport.csv")
# Assigning the data frame DublinAirport to the class
class(CorkAirport) <- c('WeatherData', 'data.frame')

summary.WeatherData(CorkAirport)
```

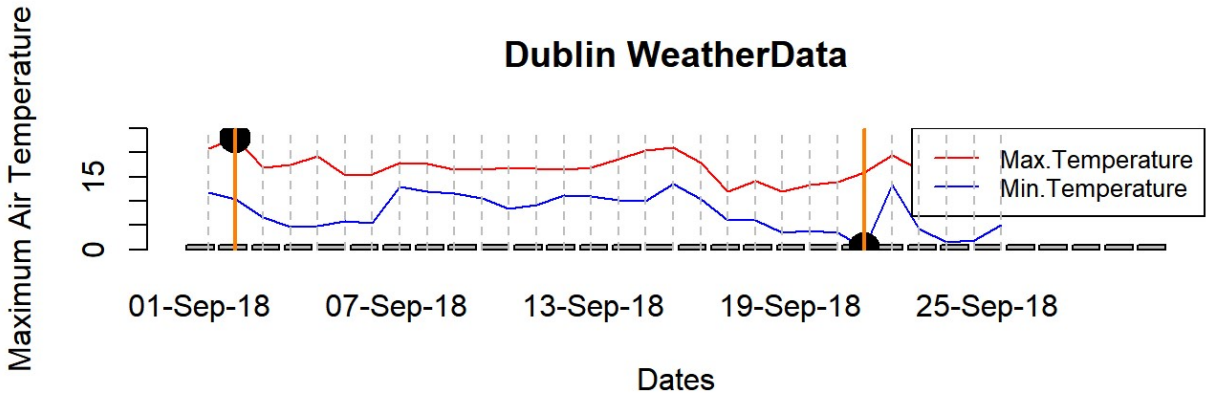
```
## Mean of Rain 2.58
## Standart deviation of Rain 4.575075
## Max of Rain 19.2
## Min of Rain 0
## Mean of maxtp 15.95
## Standart deviation of Maxtp 2.381212
## Max of Maxtp 20.2
## Min of Maxtp 10.3
## Mean of Mintp 8.686667
## Standart deviation of Mintp 2.336625
## Max of Mintp 13
## Min of Mintp 4.7
```

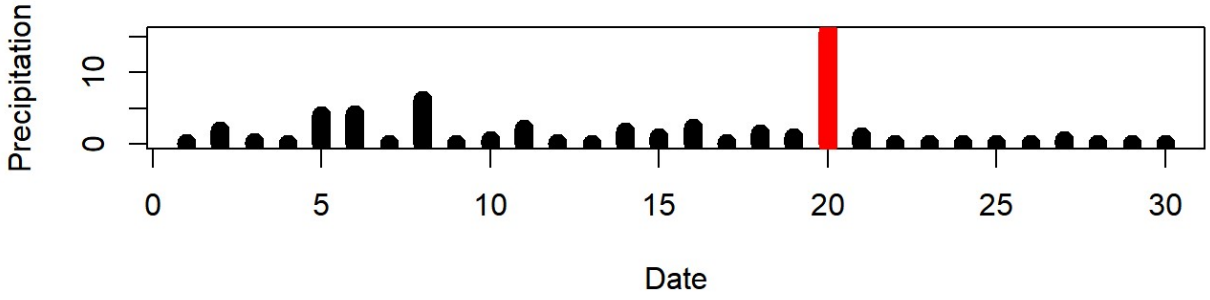
Interpreting both Dublin and Cork dataset, its evident that CorkAirport overweighs that DublinAirport with respect to the precipitation amount based on mean, standard deviationg, minimum, maximum and viceversa with respect to Maximum Air Temperature. For minimum air temperature, standart deviation and maximum of DublinAirport is greater than CorkAirport dataset, on the other hand, mean and minimum of CorkAirport leads DublinAirport dataset.

4. Create an S3 plot method for the class WeatherData that produces the following plots.

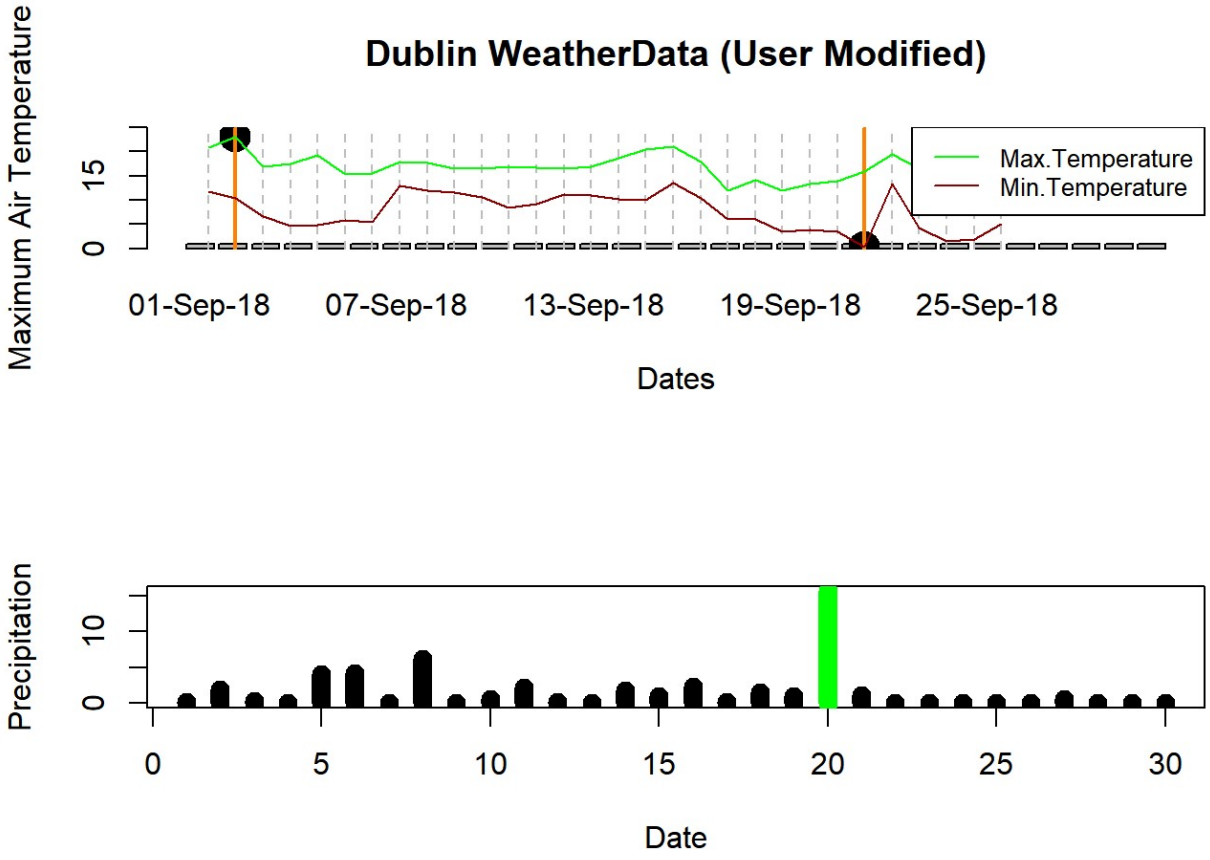
```
plot.WeatherData <- function(object, user, color1, color2){
  # For printing the plot in single panel
  par(mfrow=c(2,1))
  if(all(object == DublinAirport) & missing(user)){
    #For ploting the axis with limits
    plot(object$ï..date, xlab = "Dates", ylab = "Maximum Air Temperature", main = "Dub
lin WeatherData", ylim=c(0,25))
  }
  else if(all(object == CorkAirport) & missing(user)){
    #For ploting the axis with limits
    plot(object$ï..date, xlab = "Dates", ylab = "Maximum Air Temperature", main = "Cor
k WeatherData", ylim=c(0,25))
  }
  else if(all(object == DublinAirport) & user==TRUE){
    #For ploting the axis with limits
    plot(object$ï..date, xlab = "Dates", ylab = "Maximum Air Temperature", main = "Dub
lin WeatherData (User Modified)", ylim=c(0,25))
  }
  else if(all(object == CorkAirport) & user==TRUE){
    #For ploting the axis with limits
    plot(object$ï..date, xlab = "Dates", ylab = "Maximum Air Temperature", main = "Cor
k WeatherData (User Modified)", ylim=c(0,25))
  }
  #For generating the Maximum temperature plot using red line
  lines(object$maxtp, col="red")
  #For generating the Minimum temperature plot using blue line
  lines(object$mintp, col="blue")

  #To mention the legend of the plot
  legend("topright", legend=c("Max.Temperature", "Min.Temperature"), col=c("red", "blu
e"), lty=1, cex=0.8)
  # For printing the vertical line
  abline(v=object$ï..date, lty=2, col="grey")
  #Hightlighting the highest maximum temperature registered
  points(object[which.max(object$maxtp),]$ï..date,object[which.max(object$maxtp),]$max
tp, pch=19, bg="black",col="black", lwd=10)
  #Hightlighting the highest maximum temperature registered using vertical line
  abline(v=object[which.max(object$maxtp),]$ï..date ,col='darkorange1', lwd=2)
  #Hightlighting the lowest minimum temperature registered
  points(object[which.min(object$mintp),]$ï..date,object[which.min(object$mintp),]$min
tp, pch=19, bg="black", col="black", lwd=10)
  #Hightlighting the lowest minimum temperature registered using vertical line
  abline(v=object[which.min(object$mintp),]$ï..date ,col="darkorange1", lwd=2)

  # USer control to change the color
  if(!missing(user)){
    #For generating the Maximum temperature plot using user input color1
    lines(object$maxtp, col=color1)
```

```
    #For generating the Minimum temperature plot using user input color2
    lines(object$mintp, col=color2)
    #To mention the legend of the plot as per user's input
    legend("topright", legend=c("Max.Temperature", "Min.Temperature"), col=c(color1, c
olor2), lty=1, cex=0.8)
  }

  #Bottom plot
  # Ploting the Precipation plot with vertical line
  plot(as.numeric(object$ï..date), object$rain, xlab="Date", ylab="Precipitation", typ
e="h", lwd=10)
  # Highlighting the highest amount of rainfall in red
  points(as.numeric(object[which.max(object$rain),]$ï..date),object[which.max(object$r
ain),]$rain, type="h", bg="red",col="red", lwd=10)

  # USer control to change the color
  if(!missing(user)){
    points(as.numeric(object[which.max(object$rain),]$ï..date),object[which.max(object
$rain),]$rain, type="h", col=color1, lwd=10)
  }
}
plot.WeatherData(DublinAirport, )
```

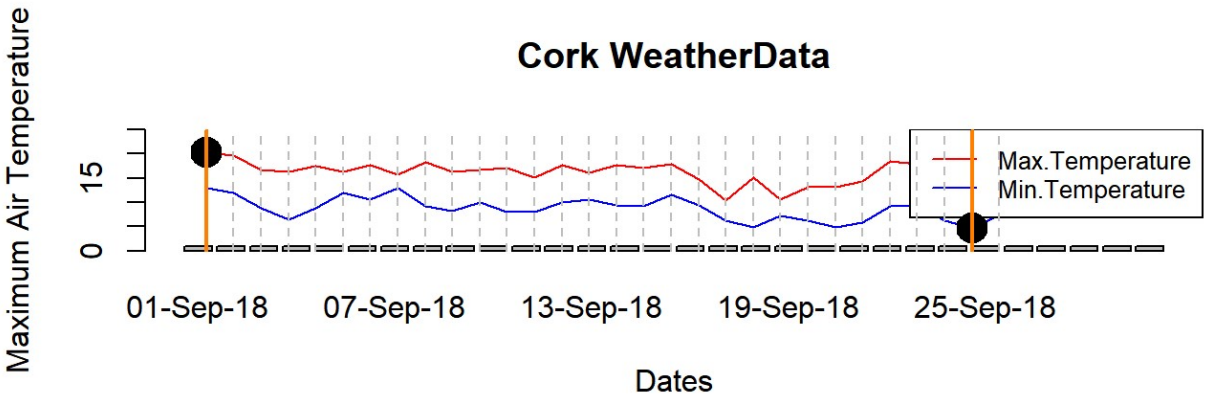


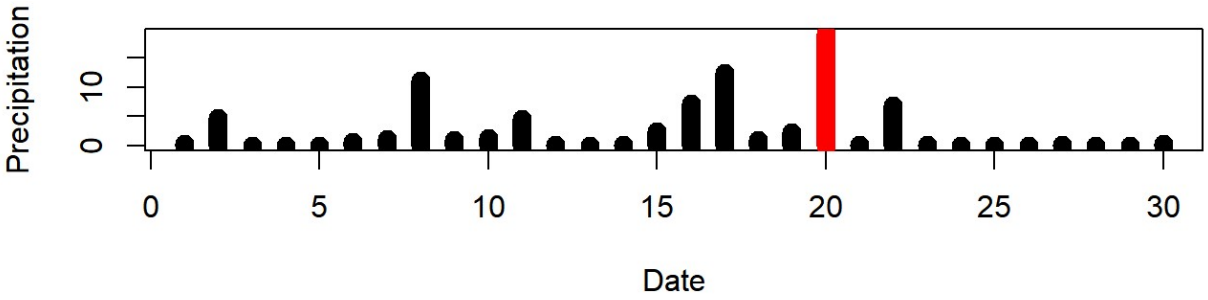


```
plot.WeatherData(DublinAirport, user=TRUE, color1 = "green", color2= "dark red")
```

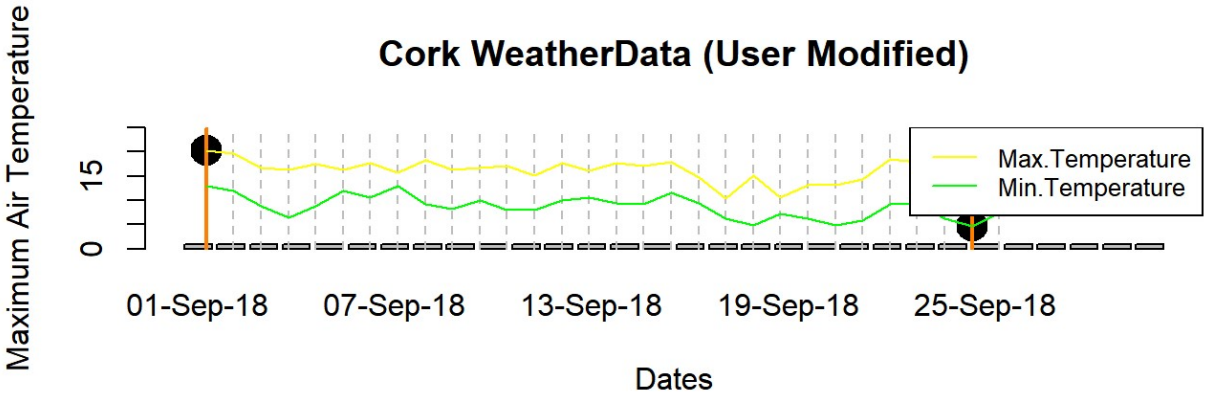


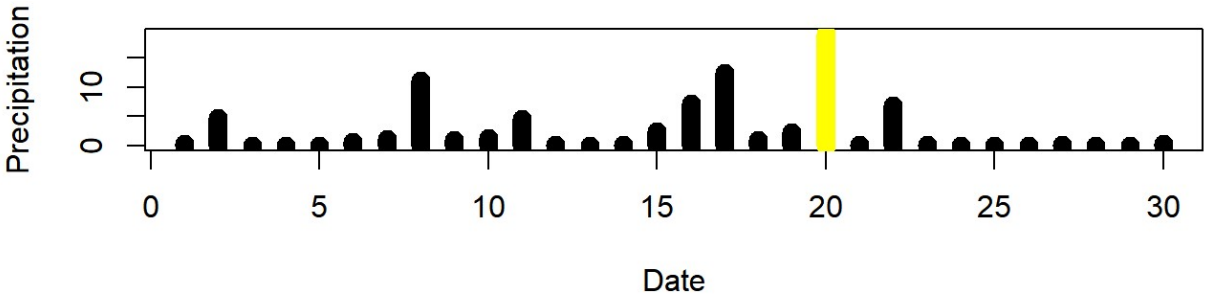

```
plot.WeatherData(CorkAirport)
```






```
plot.WeatherData(CorkAirport, user=TRUE, color1 = "yellow", color2= "green")
```


Cork WeatherData (User Modified)
Maximum Air Temperature
0
15
Max.Temperature
Min.Temperature
01-Sep-18
07-Sep-18
13-Sep-18
19-Sep-18
25-Sep-18
Dates


Precipitation
0
10
0
5
10
15
20
25
30
Date