FDA Lab-8

KHAN MOHD OWAIS RAZA
20BCD7138

Q.1] Consider the data set “airquality”.
Read first 6 lines into a new data frame “aq” (aq <- head(airquality)) and perform the following operations.
Perform the following operations using the read.table and write.table functions in R:

1. Write the data frame "aq" to a tab-separated values (TSV) file called "data.txt".
2. Read the contents of " data.txt" using the read.table function and store it in a new data frame called "data_new".
3. Display the structure and contents of "data_new" to verify that it matches the original data frame.

R Console

```
> #KHAN MOHD OWAIS RAZA
> #20BCD7138
> # Read the first 6 lines of the airquality dataset into a new data frame 'aq'
> aq <- head(airquality)
>
> # Write the data frame 'aq' to a tab-separated values (TSV) file called "data.txt"
> write.table(aq, file = "data.txt", sep = "\t", quote = FALSE, row.names = FALSE)
>
> # Read the contents of "data.txt" using the read.table function and store it in a new data frame 'data_new'
> data_new <- read.table("data.txt", sep = "\t", header = TRUE)
>
> # Display the structure and contents of 'data_new'
> cat("Structure of data_new:\n")
Structure of data_new:
> str(data_new)
'data.frame':   6 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28
 $ Solar.R: int  190 118 149 313 NA NA
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9
 $ Temp   : int  67 72 74 62 56 66
 $ Month  : int  5 5 5 5 5 5
 $ Day    : int  1 2 3 4 5 6
>
> cat("\nContents of data_new:\n")

Contents of data_new:
> print(data_new)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

Q.2] Create a new file
Line 1: Hello, World!
Line 2: How are you?
Line 3: I'm doing great!

1. Write a new file called "output.txt" with the same content as above(line1,line2,line3) using writeLines.
2. Read the contents of "output.txt" using the readLines function and store it in a variable.
3. Display the contents of the variable to verify that it matches the original text.

R Console

```
> #KHAN MOHD OWAIS RAZA
> #20BCD7138
> # Create the content
> content <- c("Hello, World!", "How are you?", "I'm doing great!")
>
> # Write the content to a new file called "output.txt" using writeLines
> writeLines(content, "output.txt")
>
> # Read the contents of "output.txt" using the readLines function and store it in a variable
> read_content <- readLines("output.txt")
>
> # Display the contents of the variable to verify that it matches the original text
> cat("Contents of read_content:\n")
Contents of read_content:
> cat(read_content, sep = "\n")
Hello, World!
How are you?
I'm doing great!
>
```

Q.3] Consider the following dataframe
students <- data.frame(
 Name = c("Alice", "Bob", "John", "Jane"),
 Age = c(25, 30, 35, 40),
 Grade = c("A", "B", "A", "B")
)
Perform the following operations using the read.delim and write.delim functions in R:

1. Write the data frame "students" to a tab-separated values (TSV) file called "students.tsv".
2. Read the contents of "students.tsv" using the read.delim function and store it in a new data frame called "students_new".
3. Display the structure and contents of "students_new" to verify that it matches the original data frame.

```
> students <- data.frame(
+     Name = c("Alice", "Bob", "John", "Jane"),
+     Age = c(25, 30, 35, 40),
+     Grade = c("A", "B", "A", "B")
+ )
>
> students
   Name Age Grade
1 Alice  25     A
2   Bob  30     B
3  John  35     A
4  Jane  40     B
>
```

```
> # Create the dataframe "students"
> students <- data.frame(
+     Name = c("Alice", "Bob", "John", "Jane"),
+     Age = c(25, 30, 35, 40),
+     Grade = c("A", "B", "A", "B")
+ )
>
> # Write the data frame "students" to a TSV file called "students.tsv"
> write.table(students, "students.tsv", sep = "\t", row.names = FALSE)
>
> # Read the contents of "students.tsv" into "students_new"
> students_new <- read.table("students.tsv", header = TRUE, sep = "\t")
>
> # Display the structure and contents of "students_new"
> str(students_new)
'data.frame':   4 obs. of  3 variables:
 $ Name : chr  "Alice" "Bob" "John" "Jane"
 $ Age  : int  25 30 35 40
 $ Grade: chr  "A" "B" "A" "B"
> head(students_new)
   Name Age Grade
1 Alice  25     A
2   Bob  30     B
3  John  35     A
4  Jane  40     B
>
```

Q.4] Consider the following dataframe
students <- data.frame(
  Name = c("John", "Jane", "Alice", "Bob"),
  Math = c(90, 85, 92, 88),
  Science = c(95, 88, 90, 94),
  English = c(80, 92, 85, 88)
)
Perform the following operations using the **read.csv** and **write.csv** functions in R:

1. Write the data frame "students" to a comma-separated values (CSV) file called "students_data.csv".
2. Read the contents of "students_data.csv" using the **read.csv** function and store it in a new data frame called "students_new".
3. Display the structure and contents of "students_new" to verify that it matches the original data frame.

R Console

```
> #KHAN MOHD OWAIS RAZA
> #20BCD7138
> # Create the students data frame
> students <- data.frame(
+   Name = c("John", "Jane", "Alice", "Bob"),
+   Math = c(90, 85, 92, 88),
+   Science = c(95, 88, 90, 94),
+   English = c(80, 92, 85, 88)
+ )
>
> # Write the data frame "students" to a comma-separated values (CSV) file called "students_data.csv"
> write.csv(students, "students_data.csv", row.names = FALSE)
>
> # Read the contents of "students_data.csv" using the read.csv function and store it in a new data frame called "students_new"
> students_new <- read.csv("students_data.csv")
>
> # Display the structure and contents of "students_new" to verify that it matches the original data frame
> cat("Structure of students_new:\n")
Structure of students_new:
> str(students_new)
'data.frame':   4 obs. of  4 variables:
 $ Name   : chr  "John" "Jane" "Alice" "Bob"
 $ Math   : int  90 85 92 88
 $ Science: int  95 88 90 94
 $ English: int  80 92 85 88
>
> cat("\nContents of students_new:\n")

Contents of students_new:
> print(students_new)
   Name Math Science English
1  John   90      95      80
2  Jane   85      88      92
3 Alice   92      90      85
4   Bob   88      94      88
```

Q.5] Consider a data frame named "students" that contains information about students' grades. The data frame has the following structure:

```
students <- data.frame(
  Name = c("John", "Jane", "Alice", "Bob"),
  Math = c(90, 85, 92, 88),
  Science = c(95, 88, 90, 94),
  English = c(80, 92, 85, 88)
)
```

Perform the following operations using the **save** and **load** functions in R:

1. Save the data frame "students" as a binary file called "students_data.RData".
2. Clear the current environment to remove the "students" data frame.
3. Load the "students_data.RData" file into the R environment.
4. Display the structure and contents of the loaded "students" data frame to verify that it matches the original data frame.

R Console

```
> #KHAN MOHD OWAIS RAZA
> #20BCD7138
> # Create the students data frame
> students <- data.frame(
+   Name = c("John", "Jane", "Alice", "Bob"),
+   Math = c(90, 85, 92, 88),
+   Science = c(95, 88, 90, 94),
+   English = c(80, 92, 85, 88)
+ )
>
> # Save the data frame "students" as a binary file called "students_data.RData"
> save(students, file = "students_data.RData")
>
> # Clear the current environment to remove the "students" data frame
> rm(list = ls())
>
> # Load the "students_data.RData" file into the R environment
> load("students_data.RData")
>
> # Display the structure and contents of the loaded "students" data frame to verify that it matches the original data frame
> cat("Structure of students:\n")
Structure of students:
> str(students)
'data.frame':   4 obs. of  4 variables:
 $ Name   : chr  "John" "Jane" "Alice" "Bob"
 $ Math   : num  90 85 92 88
 $ Science: num  95 88 90 94
 $ English: num  80 92 85 88
>
> cat("\nContents of students:\n")

Contents of students:
> print(students)
   Name Math Science English
1  John   90      95      80
2  Jane   85      88      92
3 Alice   92      90      85
4   Bob   88      94      88
```

Q.6] Create a file a text file named "sales.txt" that contains the monthly sales data for a company. Each line in the file represents the sales for a specific month. The file has the following format:
January,10000
February,15000
March,12000
April,18000
Perform the following operations using the **scan** function in R:

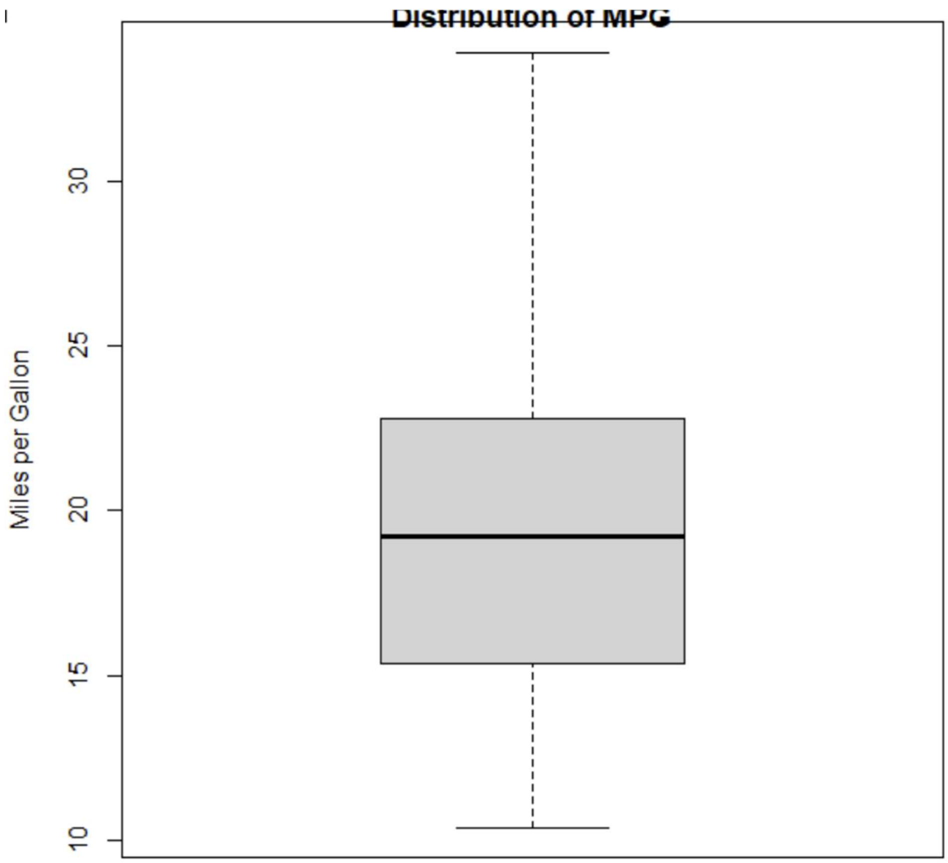
1. Read the "sales.txt" file and store the data in a vector.
2. Calculate the total sales for all the months.
3. Display the total sales.

```
#KHAN MOHD OWAIS RAZA
#20BCD7138
# Read the "sales.txt" file and store the data in a
vector
sales <- scan("sales.txt", what = list("", 0), sep = ",",
strip.white = TRUE)
# Extract the sales values from the scanned data
sales <- unlist(sales[[2]])
# Calculate the total sales for all the months
total_sales <- sum(sales)
# Display the total sales
total_sales
```

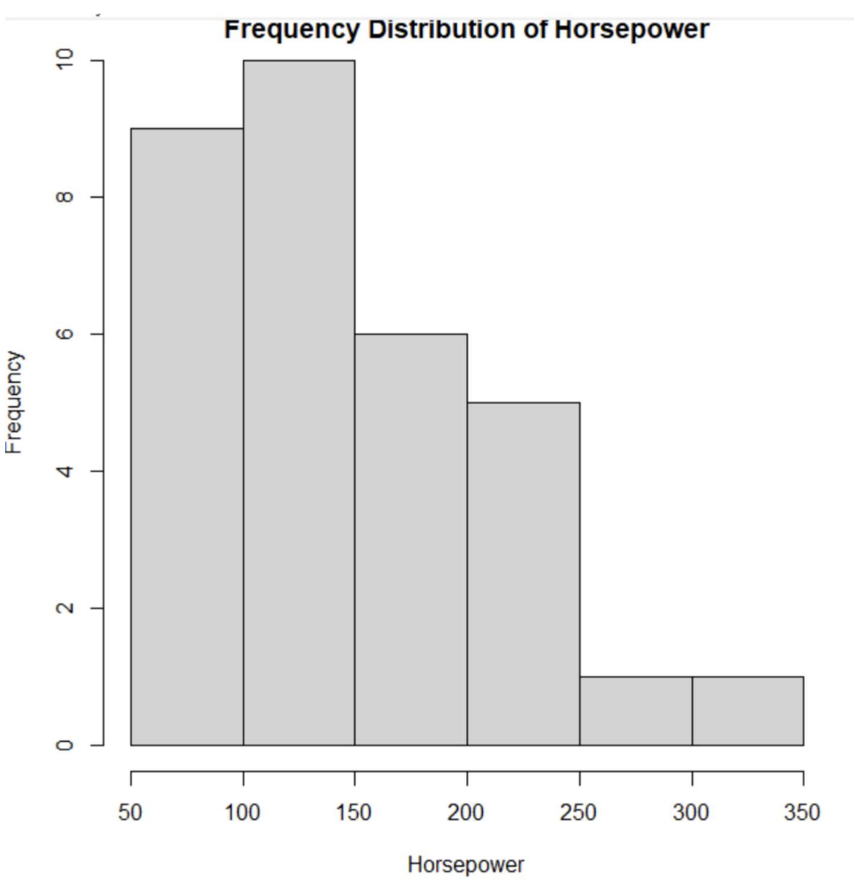
Q.7] Consider mtcars dataset Perform the following operations to visualize the data using different plot types:

1. Create a boxplot to visualize the distribution of the "mpg" variable (miles per gallon) in the "mtcars" dataset.
2. Generate a histogram to examine the frequency distribution of the "hp" variable (horsepower) in the dataset.
3. Create a pie chart to display the proportion of different car cylinders ("cyl") in the dataset.
4. Generate a line graph to visualize the trend of the "wt" variable (weight) across different car models.
5. Create a scatter plot to analyze the relationship between the "mpg" variable and the "hp" variable.
6. Generate a bar plot to compare the average "mpg" values for different car models.

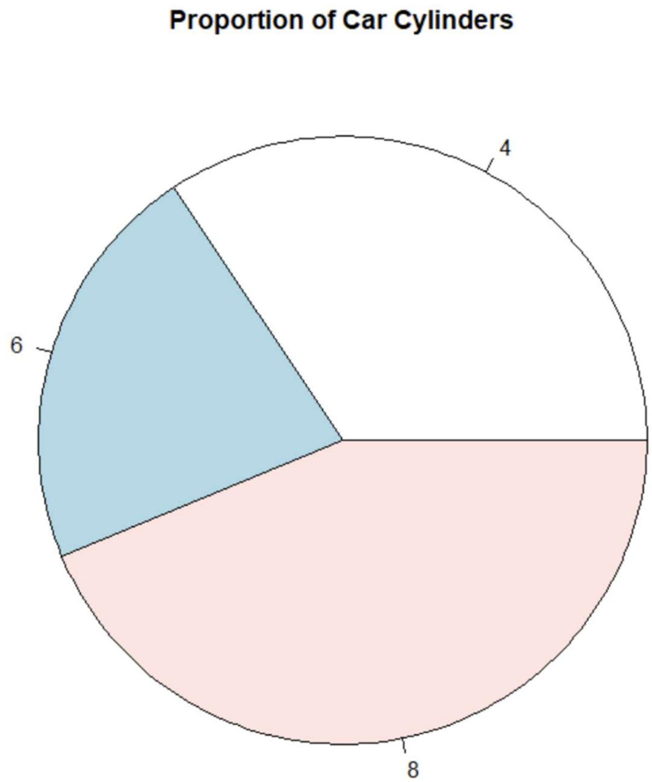
```
> # Open a new plotting window with adjusted size
> dev.new(width = 10, height = 6)
NULL
>
> # Plotting code goes here
> plot(mtcars$mpg, main = "Distribution of MPG", ylab = "Miles per Gallon")
>
> par(mar = c(5, 4, 0.5, 2))  # Adjust the margin
> boxplot(mtcars$mpg, main = "Distribution of MPG", ylab = "Miles per Gallon")
>
```



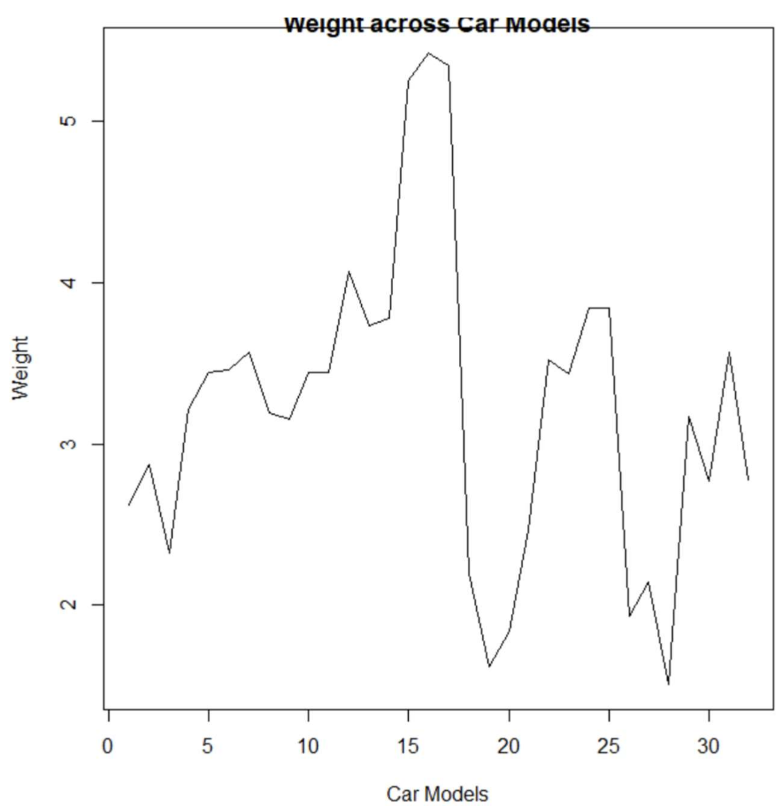

```
> par(mar = c(5, 4, 0.5, 2))
> hist(mtcars$hp, main = "Frequency Distribution of Horsepower", xlab = "Horsepower")
>
> |
```



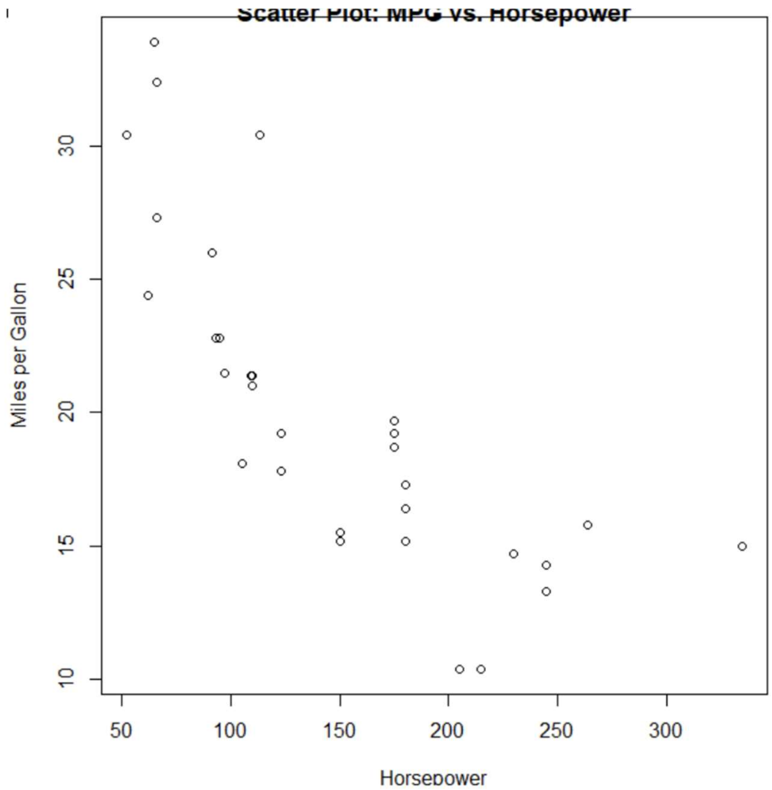
```
> par(mar = c(1, 1, 1, 1))
> cyl_counts <- table(mtcars$cyl)
> pie(cyl_counts, main = "Proportion of Car Cylinders")
>
>
```



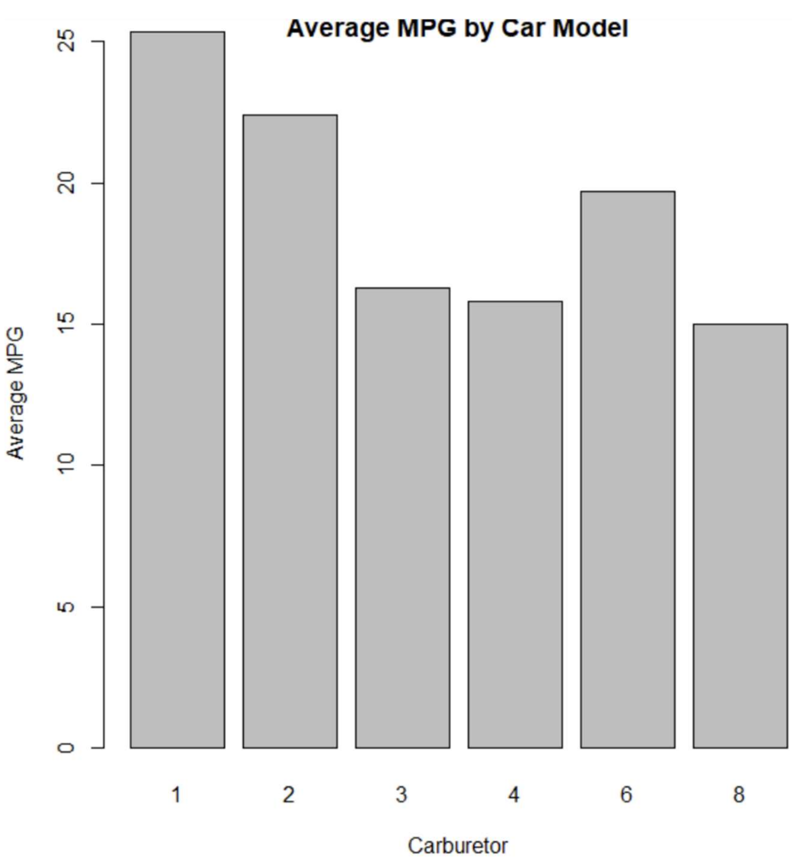

```
> par(mar = c(5, 4, 0.5, 2))
> plot(mtcars$wt, type = "l", main = "Weight across Car Models", xlab = "Car Models", ylab = "Weight")
>
```



```
> par(mar = c(5, 4, 0.5, 2))
> plot(mtcars$hp, mtcars$mpg, main = "Scatter Plot: MPG vs. Horsepower", xlab = "Horsepower", ylab = "Miles per Gallon")
>
```



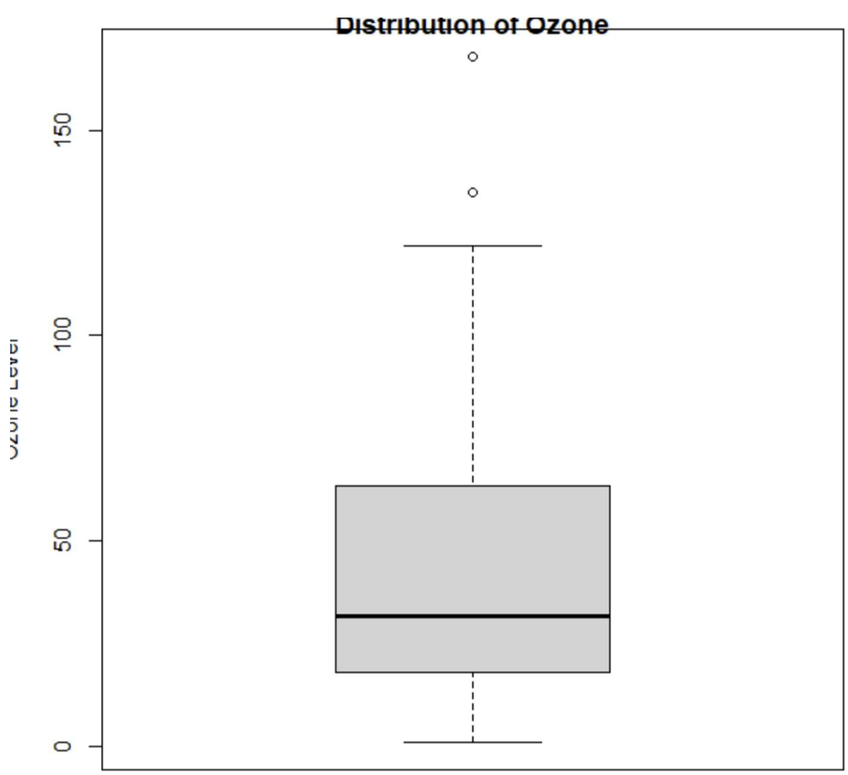

```
> par(mar = c(5, 4, 0.5, 2))
> barplot(tapply(mtcars$mpg, mtcars$carb, mean), main = "Average MPG by Car Model", xlab = "Carburetor", ylab = "Average MPG")
>
>
```



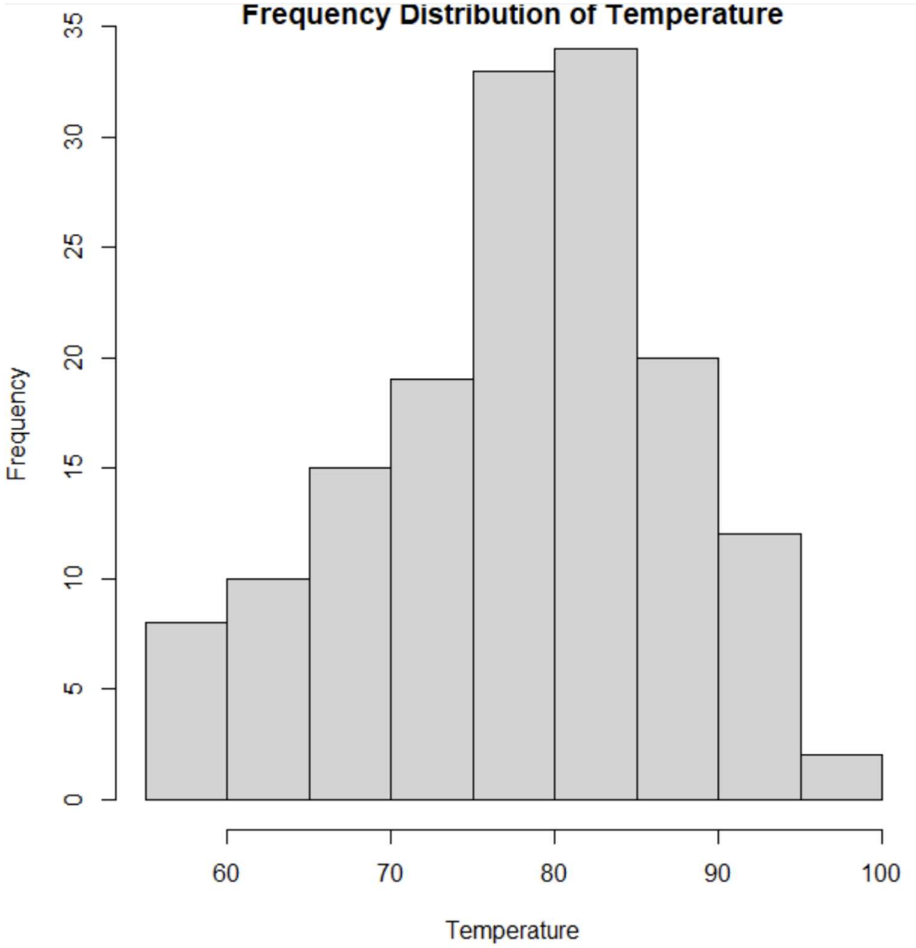
Q.8] Consider airquality dataset Perform the following operations to visualize the data using different plot types:

1. Create a boxplot to visualize the distribution of ozone levels ("Ozone") in the "airquality" dataset.
2. Generate a histogram to examine the frequency distribution of temperatures ("Temp") in the dataset.
3. Create a pie chart to display the proportion of different wind directions ("Wind") in the dataset.
4. Generate a line graph to visualize the variation of solar radiation ("Solar.R") over time.
5. Create a scatter plot to analyze the relationship between ozone levels ("Ozone") and wind speeds ("Wind.Speed").
6. Generate a bar plot to compare the average temperatures ("Temp") for different months.

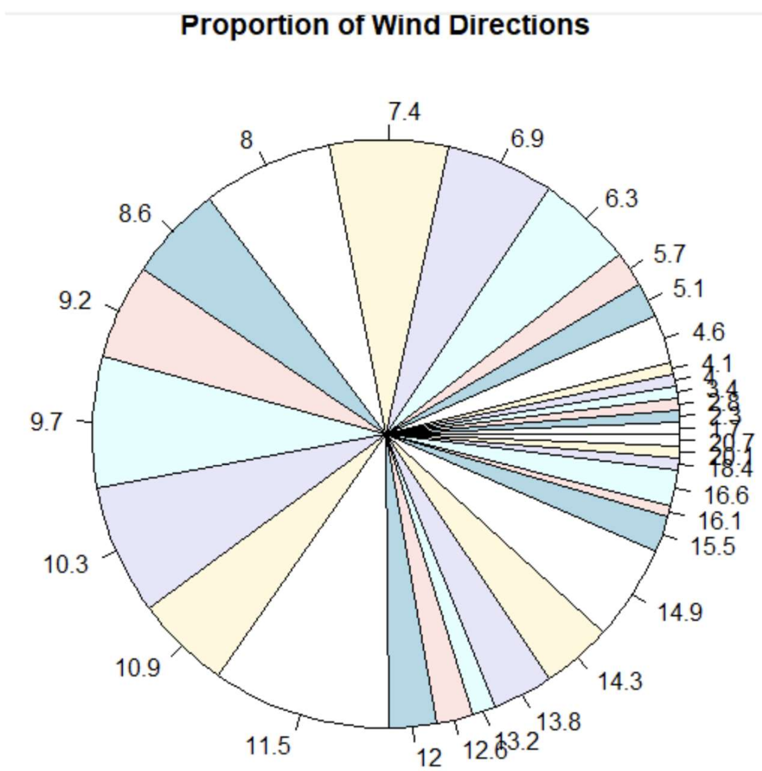
```
> boxplot(airquality$Ozone, main = "Distribution of Ozone", ylab = "Ozone Level")
```



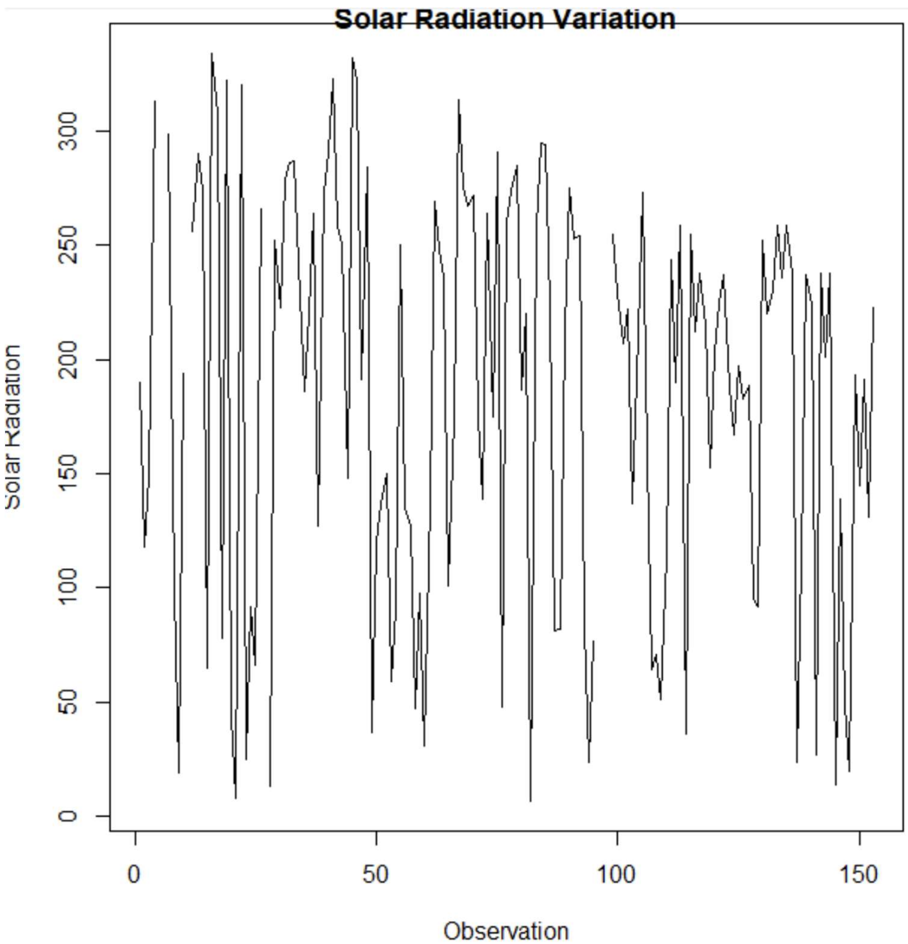
```
> hist(airquality$Temp, main = "Frequency Distribution of Temperature", xlab = "Temperature")
>
```




```
> wind_counts <- table(airquality$Wind)
> pie(wind_counts, main = "Proportion of Wind Directions")
```



```
> plot(airquality$Solar.R, type = "l", main = "Solar Radiation Variation", xlab = "Observation", ylab = "Solar Radiation")
>
```




```
> # Identify complete cases
> complete_cases <- complete.cases(airquality$Wind.Speed, airquality$Ozone)
>
> # Subset the dataset to complete cases
> subset_data <- airquality[complete_cases, ]
>
> # Create scatter plot
> plot(subset_data$Wind.Speed, subset_data$Ozone, main = "Scatter Plot: Ozone vs. Wind Speed", xlab = "Wind Speed", ylab = "Ozone Level")
```

```
> months <- factor(airquality$Month, levels = 5:9, labels = c("May", "June", "July", "August", "September"))
> barplot(tapply(airquality$Temp, months, mean), main = "Average Temperature by Month", xlab = "Month", ylab = "Average Temperature")
>
```