# DAY-2

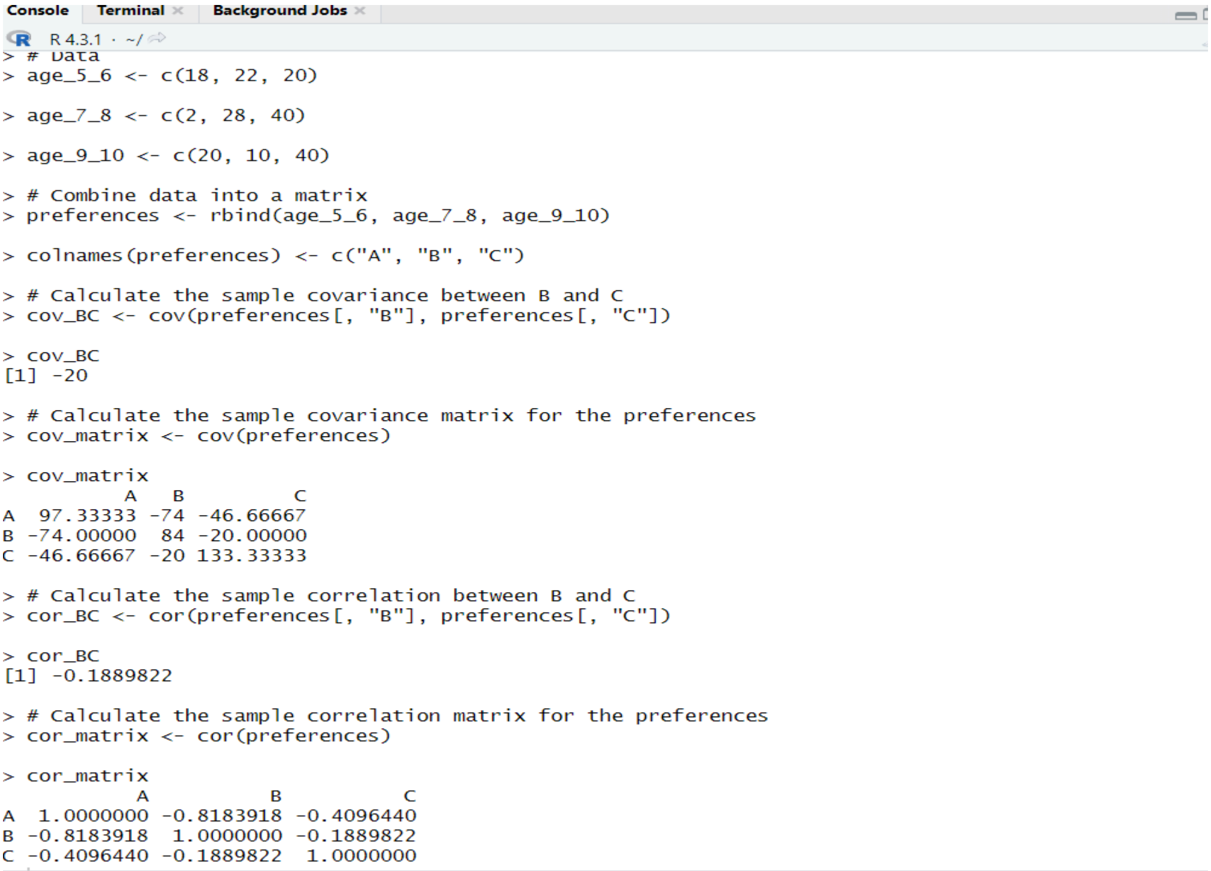
1.Covariance and correlation Children of three ages are asked to indicate their preference for three photographs of adults. Do the data suggest that there is a significant relationship between age and photograph preference? What is wrong with this study? Photograph: Age of child A B C 5-6 years: 18 22 20 7-8 years: 2 28 40 9-10 years: 20 10 40

1.Use cov() to calculate the sample covariance between B and C.
2. Use another call to cov() to calculate the sample covariance matrix for the preferences.
3. Use cor() to calculate the sample correlation between B and C.
4. Use another call to cor() to calculate the sample correlation matrix for the

Code:

```
# Data
age_5_6 <- c(18, 22, 20)
age_7_8 <- c(2, 28, 40)
age_9_10 <- c(20, 10, 40)

# Combine data into a matrix
preferences <- rbind(age_5_6, age_7_8, age_9_10)
colnames(preferences) <- c("A", "B", "C")

# Calculate the sample covariance between B and C
cov_BC <- cov(preferences[, "B"], preferences[, "C"])
cov_BC

# Calculate the sample covariance matrix for the preferences
cov_matrix <- cov(preferences)
Cov_matrix

# Calculate the sample correlation between B and C
cor_BC <- cor(preferences[, "B"], preferences[, "C"])
cor_BC

# Calculate the sample correlation matrix for the preferences
cor_matrix <- cor(preferences)
cor_matrix
```

Output:



```
> # Data
> age_5_6 <- c(18, 22, 20)

> age_7_8 <- c(2, 28, 40)

> age_9_10 <- c(20, 10, 40)

> # Combine data into a matrix
> preferences <- rbind(age_5_6, age_7_8, age_9_10)

> colnames(preferences) <- c("A", "B", "C")

> # Calculate the sample covariance between B and C
> cov_BC <- cov(preferences[, "B"], preferences[, "C"])

> cov_BC
[1] -20

> # Calculate the sample covariance matrix for the preferences
> cov_matrix <- cov(preferences)

> cov_matrix
          A   B         C
A  97.33333 -74 -46.66667
B -74.00000  84 -20.00000
C -46.66667 -20 133.33333

> # Calculate the sample correlation between B and C
> cor_BC <- cor(preferences[, "B"], preferences[, "C"])

> cor_BC
[1] -0.1889822

> # Calculate the sample correlation matrix for the preferences
> cor_matrix <- cor(preferences)

> cor_matrix
           A          B          C
A  1.0000000 -0.8183918 -0.4096440
B -0.8183918  1.0000000 -0.1889822
C -0.4096440 -0.1889822  1.0000000
```

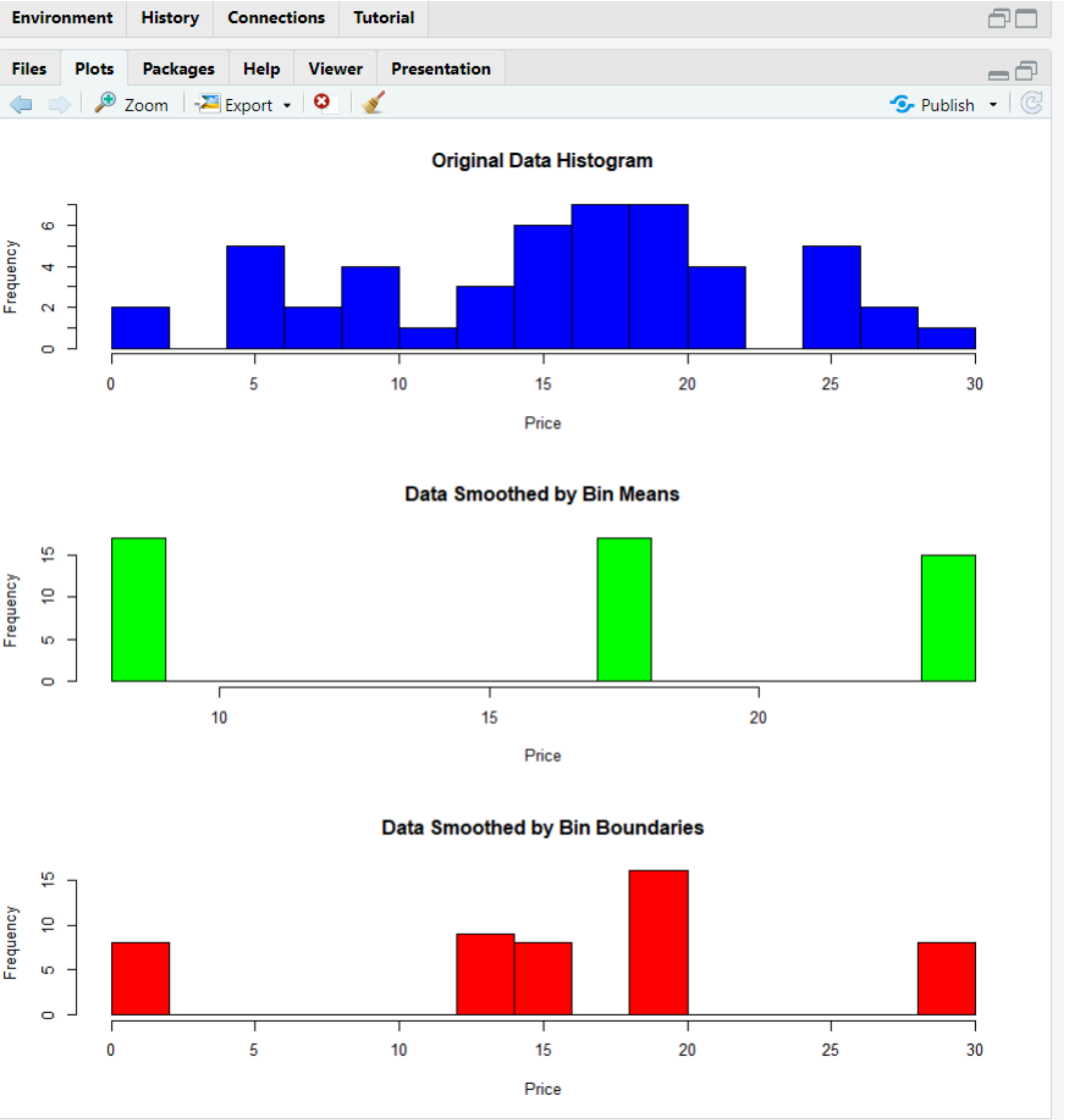
2.Imagine that you have selected data from the All Electronics data warehouse for analysis. The data set will be huge! The following data are a list of All Electronics prices for commonly sold items (rounded to the nearest dollar). The numbers have been sorted: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 8, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30 the dataset using an equal-frequency partitioning method with bin equal to 3 (ii) apply data smoothing using bin means and bin boundary. (iii) Plot Histogram for the above frequency division

Code:

```
# Data
data <- c(1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18,
18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30)

# Number of bins
k <- 3

# Equal-frequency partitioning
n <- length(data)
bin_size <- ceiling(n / k)
bins <- split(data, ceiling(seq_along(data) / bin_size))

# Smoothing by bin means
bin_means <- sapply(bins, mean)
```

```
smoothed_by_means <- unlist(lapply(1:length(bins), function(i) rep(bin_means[i],
length(bins[[i]]))))

# Smoothing by bin boundaries
smoothed_by_boundaries <- unlist(lapply(bins, function(bin) {
  c(rep(min(bin), floor(length(bin) / 2)), rep(max(bin), ceiling(length(bin) / 2)))
}))

# Plotting histograms
par(mfrow=c(3, 1))

# Original data histogram
hist(data, breaks=15, main='Original Data Histogram', xlab='Price', ylab='Frequency',
col='blue')

# Smoothed by bin means histogram
hist(smoothed_by_means, breaks=15, main='Data Smoothed by Bin Means', xlab='Price',
ylab='Frequency', col='green')

# Smoothed by bin boundaries histogram
hist(smoothed_by_boundaries, breaks=15, main='Data Smoothed by Bin Boundaries',
xlab='Price', ylab='Frequency', col='red')
```

Output:



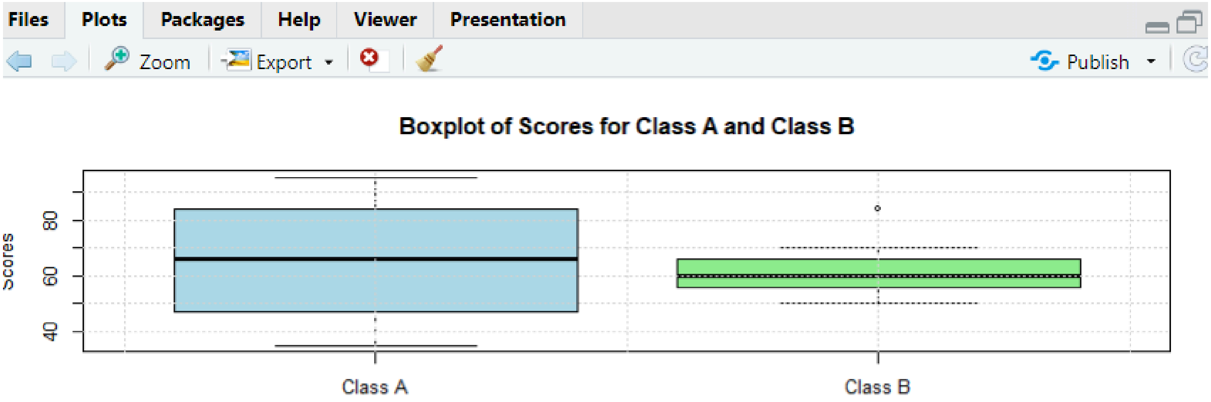
3)3.Two Maths teachers are comparing how their Year 9 classes performed in the end of year exams. Their results are as follows: Class A: 76, 35, 47, 64, 95, 66, 89, 36, 8476,35,47,64,95,66,89,36,84 Class B: 51, 56, 84, 60, 59, 70, 63, 66, 5051,56,84,60,59,70,63,66,50 (i) Find which class had scored higher mean, median and range. (ii) Plot above in boxplot and give the inferences Class B: 51, 56, 84, 60, 59, 70, 63, 66, 5051,56,84,60,59,70,63,66,50

Code:

```
# Scores for Class A and Class B
scores_A <- c(76, 35, 47, 64, 95, 66, 89, 36, 84)
scores_B <- c(51, 56, 84, 60, 59, 70, 63, 66, 50)
```

```
# Calculate mean, median, and range for Class A
mean_A <- mean(scores_A)
median_A <- median(scores_A)
range_A <- range(scores_A)
range_A_diff <- diff(range_A)

# Calculate mean, median, and range for Class B
mean_B <- mean(scores_B)
median_B <- median(scores_B)
range_B <- range(scores_B)
range_B_diff <- diff(range_B)

# Print the results
cat("Class A:\n")
cat("Mean:", mean_A, "\n")
cat("Median:", median_A, "\n")
cat("Range:", range_A_diff, "\n\n")

cat("Class B:\n")
cat("Mean:", mean_B, "\n")
cat("Median:", median_B, "\n")
cat("Range:", range_B_diff, "\n")

# Create boxplots
boxplot(scores_A, scores_B,
        names = c("Class A", "Class B"),
        main = "Boxplot of Scores for Class A and Class B",
        ylab = "Scores",
        col = c("lightblue", "lightgreen"))

# Add a grid
grid()
```

Output:

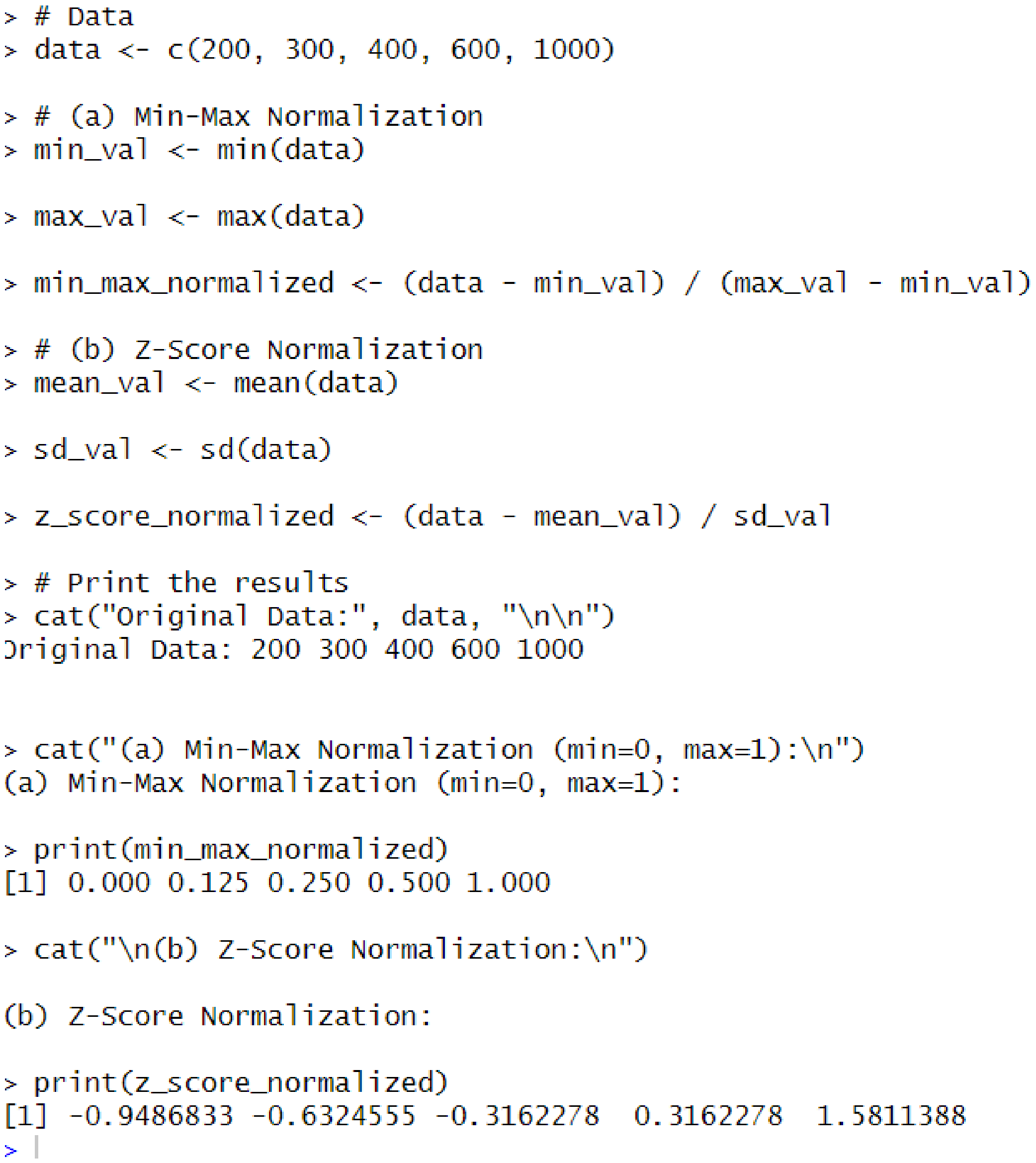


4)Let us consider one example to make the calculation method clear. Assume that the minimum and maximum values for the feature F are $50,000 and $100,000 correspondingly. It needs to range F from 0 to 1. In accordance with min-max normalization, v = $80, b) Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000 (a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization

Code:

```
# Data
data <- c(200, 300, 400, 600, 1000)

# (a) Min-Max Normalization
min_val <- min(data)
max_val <- max(data)
min_max_normalized <- (data - min_val) / (max_val - min_val)

# (b) Z-Score Normalization
mean_val <- mean(data)
sd_val <- sd(data)
z_score_normalized <- (data - mean_val) / sd_val

# Print the results
cat("Original Data:", data, "\n\n")

cat("(a) Min-Max Normalization (min=0, max=1):\n")
print(min_max_normalized)

cat("\n(b) Z-Score Normalization:\n")
print(z_score_normalized)
```

Output:



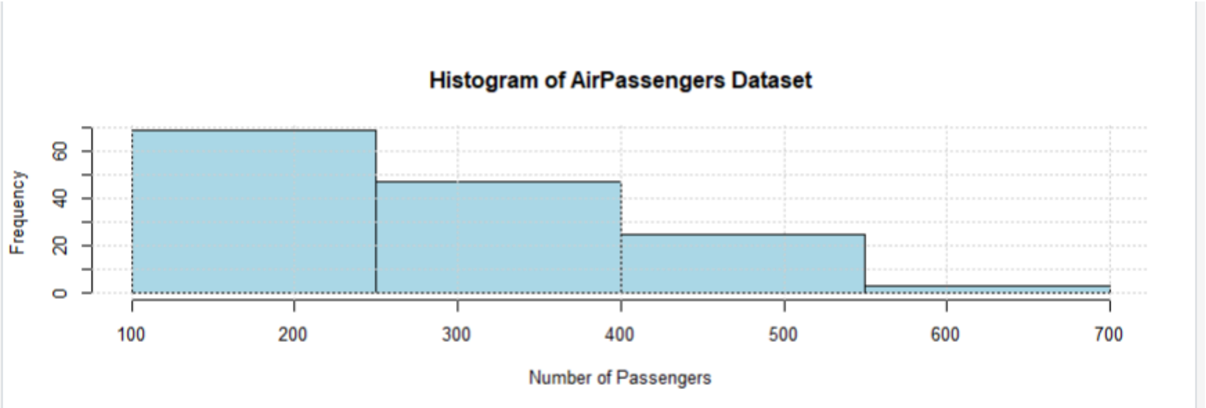
```
> # Data
> data <- c(200, 300, 400, 600, 1000)

> # (a) Min-Max Normalization
> min_val <- min(data)

> max_val <- max(data)

> min_max_normalized <- (data - min_val) / (max_val - min_val)

> # (b) Z-Score Normalization
> mean_val <- mean(data)

> sd_val <- sd(data)

> z_score_normalized <- (data - mean_val) / sd_val

> # Print the results
> cat("Original Data:", data, "\n\n")
Original Data: 200 300 400 600 1000


> cat("(a) Min-Max Normalization (min=0, max=1):\n")
(a) Min-Max Normalization (min=0, max=1):

> print(min_max_normalized)
[1] 0.000 0.125 0.250 0.500 1.000

> cat("\n(b) Z-Score Normalization:\n")

(b) Z-Score Normalization:

> print(z_score_normalized)
[1] -0.9486833 -0.6324555 -0.3162278  0.3162278  1.5811388
> 
```

5.Make a histogram for the “AirPassengers “dataset, start at 100 on the x-axis, and from values 200 to 700, make the bins 150 wide

Code:

```
# Load the AirPassengers dataset
data("AirPassengers")

# Create breaks for the histogram
breaks <- seq(100, 700, by=150)

# Create the histogram
hist(AirPassengers,
     breaks = breaks,
     main = "Histogram of AirPassengers Dataset",
     xlab = "Number of Passengers",
     ylab = "Frequency",
     col = "lightblue",
     border = "black")

# Add a grid for better visualization
grid()
```

Output:

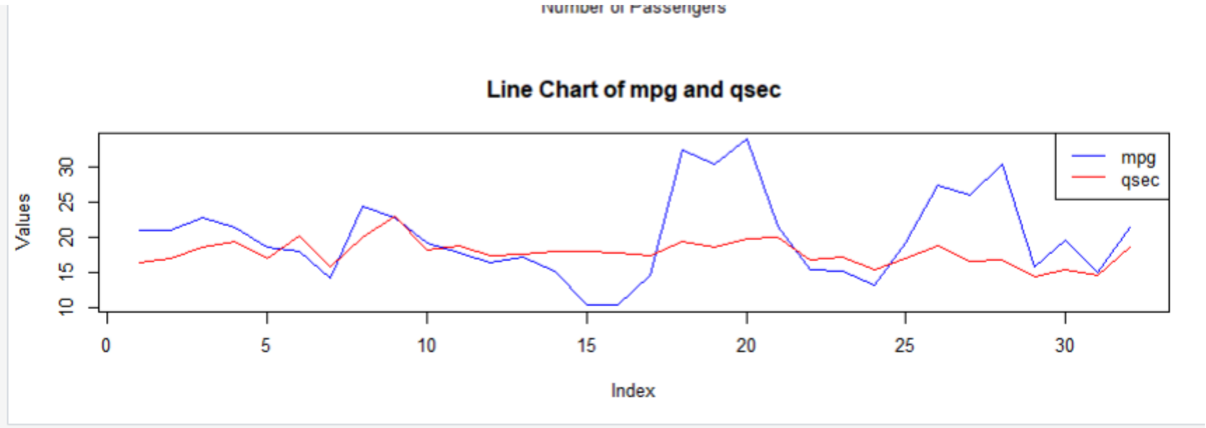


6.Obtain Multiple Lines in Line Chart using a single Plot Function in R.Use attributes“mpg”and“qsec”of the dataset “mtcars”

Code:

```
# Load the mtcars dataset
data(mtcars)

# Create a plot with 'mpg' and 'qsec' as lines
plot(mtcars$mpg, type = "l", col = "blue", ylim = range(c(mtcars$mpg, mtcars$qsec)), xlab =
"Index", ylab = "Values", main = "Line Chart of mpg and qsec")

# Add the second line for 'qsec'
lines(mtcars$qsec, col = "red")

# Add a legend
legend("topright", legend = c("mpg", "qsec"), col = c("blue", "red"), lty = 1)
```

Output:



7.Download the Dataset "water" From R dataset Link.Find out whether there is a linear relation between attributes"mortality" and"hardness" by plot function.Fit the Data into the Linear Regression model.Predict the mortality for the hardness=88.

Code:

```
# Load the 'water' dataset from the datasets package
data(water, package = "datasets")

# Check the structure of the dataset
str(water)

# Plot the relationship between 'mortality' and 'hardness'
plot(water$hardness, water$mortality,
     xlab = "Hardness", ylab = "Mortality",
     main = "Scatterplot of Mortality vs Hardness",
     pch = 16)

# Fit a linear regression model
model <- lm(mortality ~ hardness, data = water)

# Add the regression line to the plot
abline(model, col = "red")

# Display the summary of the model
summary(model)

# Predict mortality for hardness = 88
predicted_mortality <- predict(model, newdata = data.frame(hardness = 88))

# Print the predicted mortality
Predicted_mortality
```

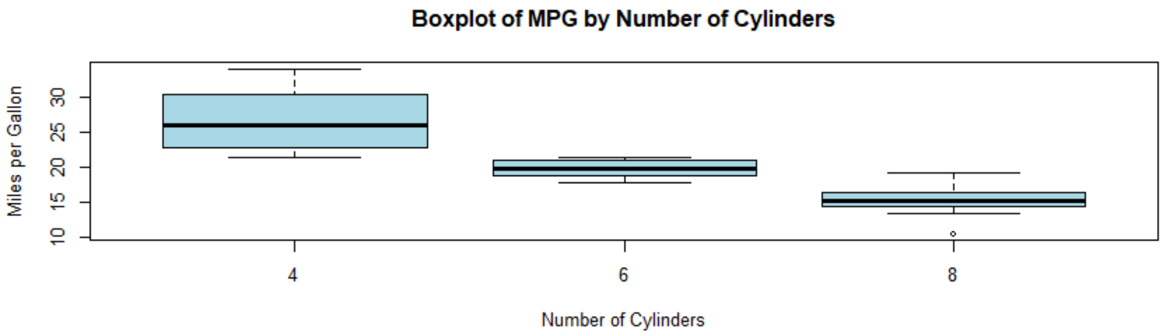
Output:

8.Create a Boxplot graph for the relation between "mpg"(miles per galloon) and "cyl"(number of Cylinders) for the dataset "mtcars" available in R Environment

Code:

```
# Load the mtcars dataset
data(mtcars)

# Create a boxplot for 'mpg' by 'cyl'
boxplot(mpg ~ cyl, data = mtcars,
        xlab = "Number of Cylinders",
        ylab = "Miles per Gallon",
        main = "Boxplot of MPG by Number of Cylinders",
        col = "lightblue")
```

Output:

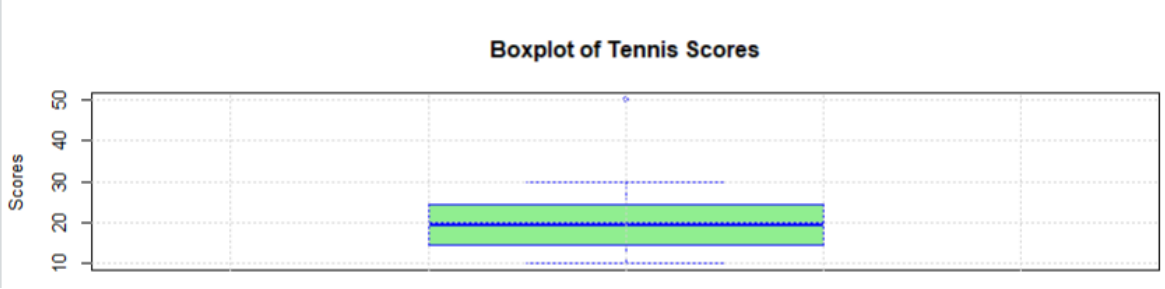


9. Assume the Tennis coach wants to determine if any of his team players are scoring outliers. To visualize the distribution of points scored by his players, then how can he decide to develop the box plot? Give suitable example using Boxplot visualization technique.

Code:

```
# Example dataset of tennis scores
scores <- c(10, 12, 15, 14, 16, 20, 22, 19, 25, 24, 30, 50)
# Create a boxplot for tennis scores
boxplot(scores,
        main = "Boxplot of Tennis Scores",
        ylab = "Scores",
        col = "lightgreen",
        border = "blue",
        horizontal = FALSE)

# Add a grid for better visualization
grid()
```

Output:

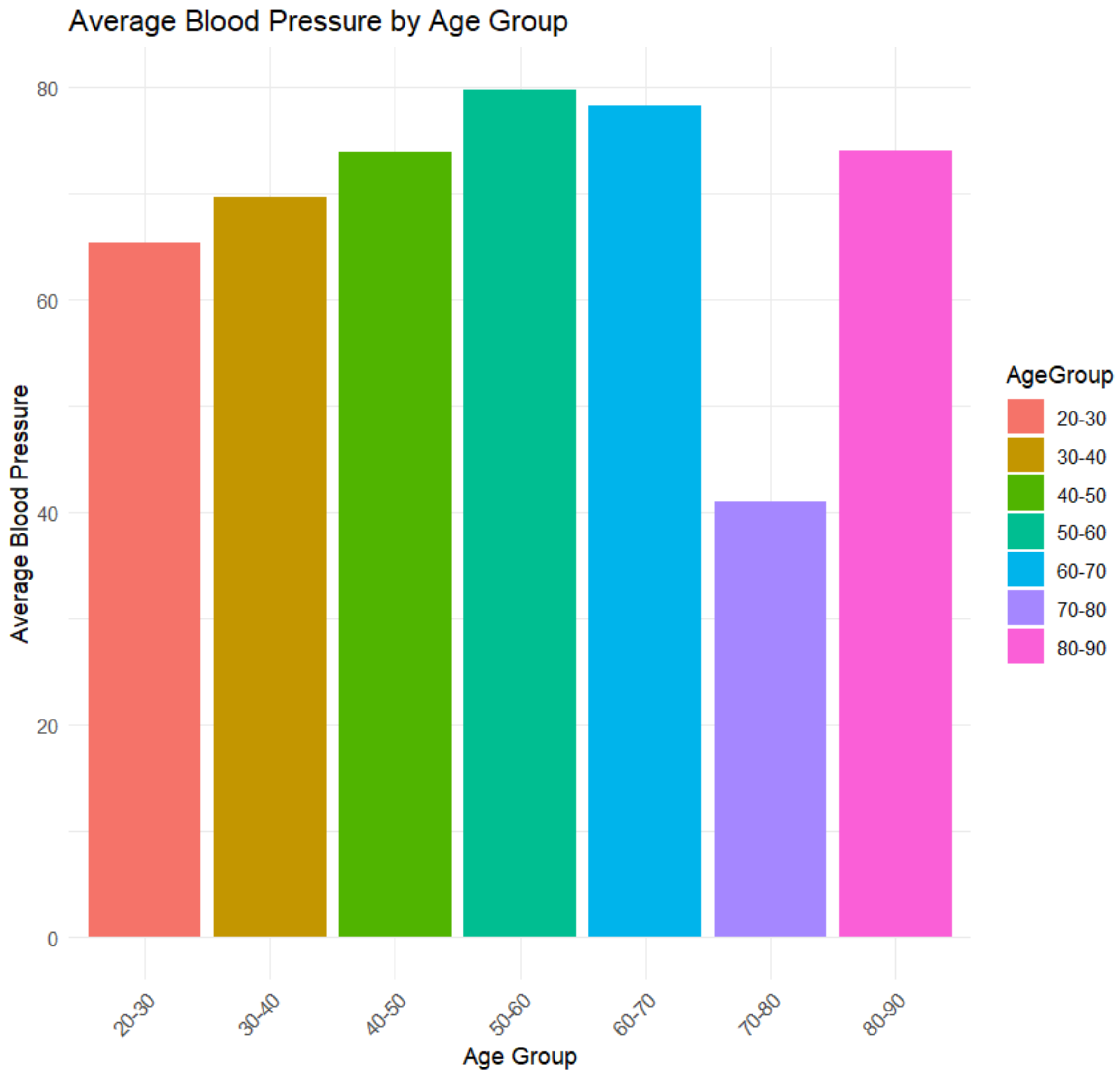


10. Implement using R language in which age group of people are affected byblood pressure based on the diabetes dataset show it using scatterplot and bar chart (that is BloodPressure vs Age using dataset “diabetes.csv”)

Code:

```
# Load necessary libraries
library(ggplot2)
library(dplyr)

# Load the dataset from a CSV file
diabetes <- read.csv("C:/Users/monis/OneDrive/Desktop/DWDM/diabetes.csv")

# Check the structure of the dataset
str(diabetes)

# Create a scatterplot of BloodPressure vs Age
ggplot(diabetes, aes(x = Age, y = BloodPressure)) +
  geom_point(color = "blue") +
  labs(title = "Scatterplot of Blood Pressure vs Age",
       x = "Age",
       y = "Blood Pressure") +
  theme_minimal()

# Define age groups
diabetes <- diabetes %>%
  mutate(AgeGroup = cut(Age, breaks = seq(0, 100, by = 10),
                        labels = paste(seq(0, 90, by = 10), seq(10, 100, by = 10), sep = "-"),
                        right = FALSE))

# Calculate average Blood Pressure for each Age Group
avg_bp_by_age_group <- diabetes %>%
  group_by(AgeGroup) %>%
  summarise(AverageBloodPressure = mean(BloodPressure, na.rm = TRUE))

# Create a bar chart of average Blood Pressure by Age Group
ggplot(avg_bp_by_age_group, aes(x = AgeGroup, y = AverageBloodPressure, fill =
AgeGroup)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Blood Pressure by Age Group",
       x = "Age Group",
       y = "Average Blood Pressure") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Output:


Average Blood Pressure by Age Group
Average Blood Pressure
Age Group
AgeGroup