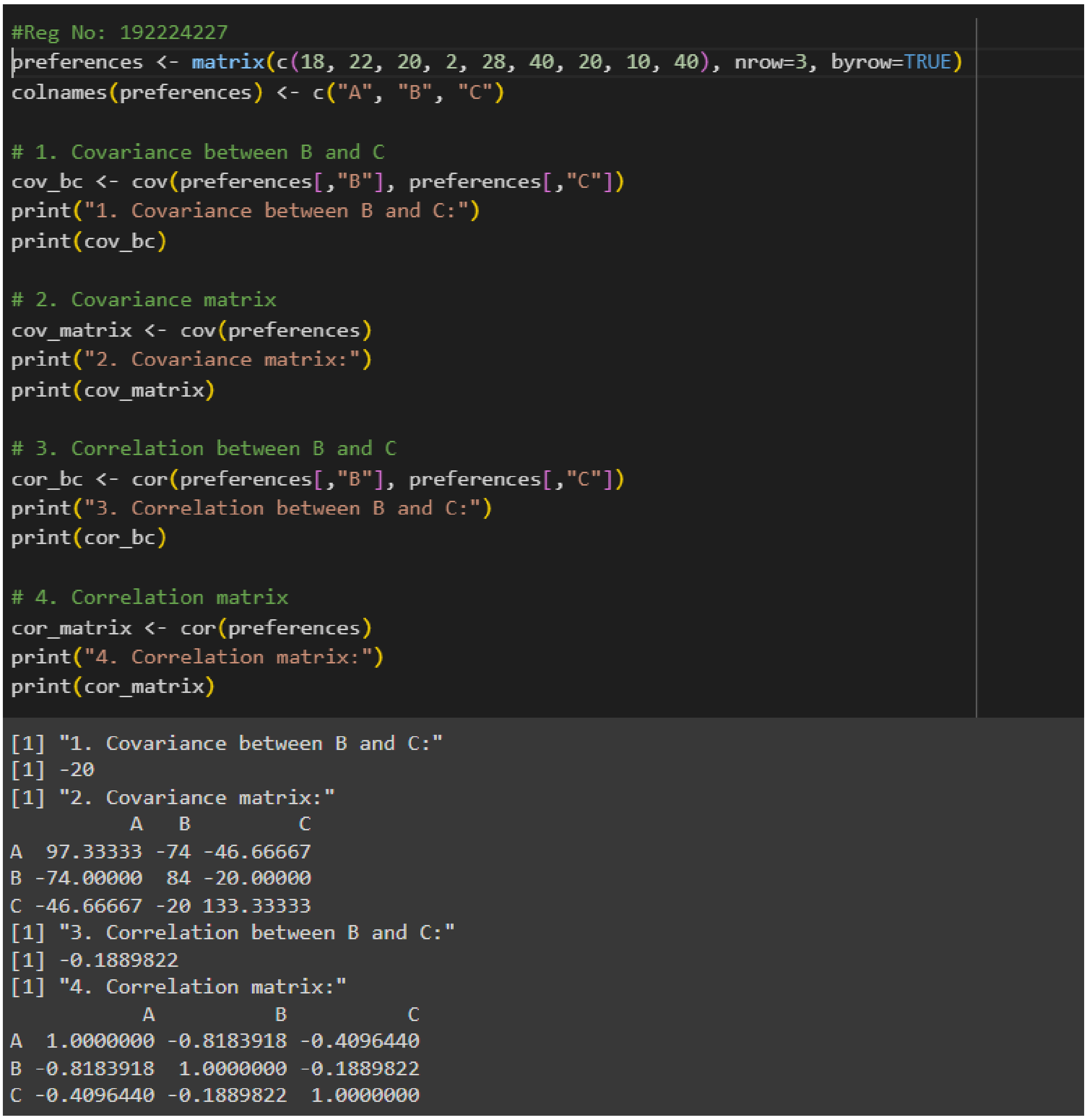
Day-2

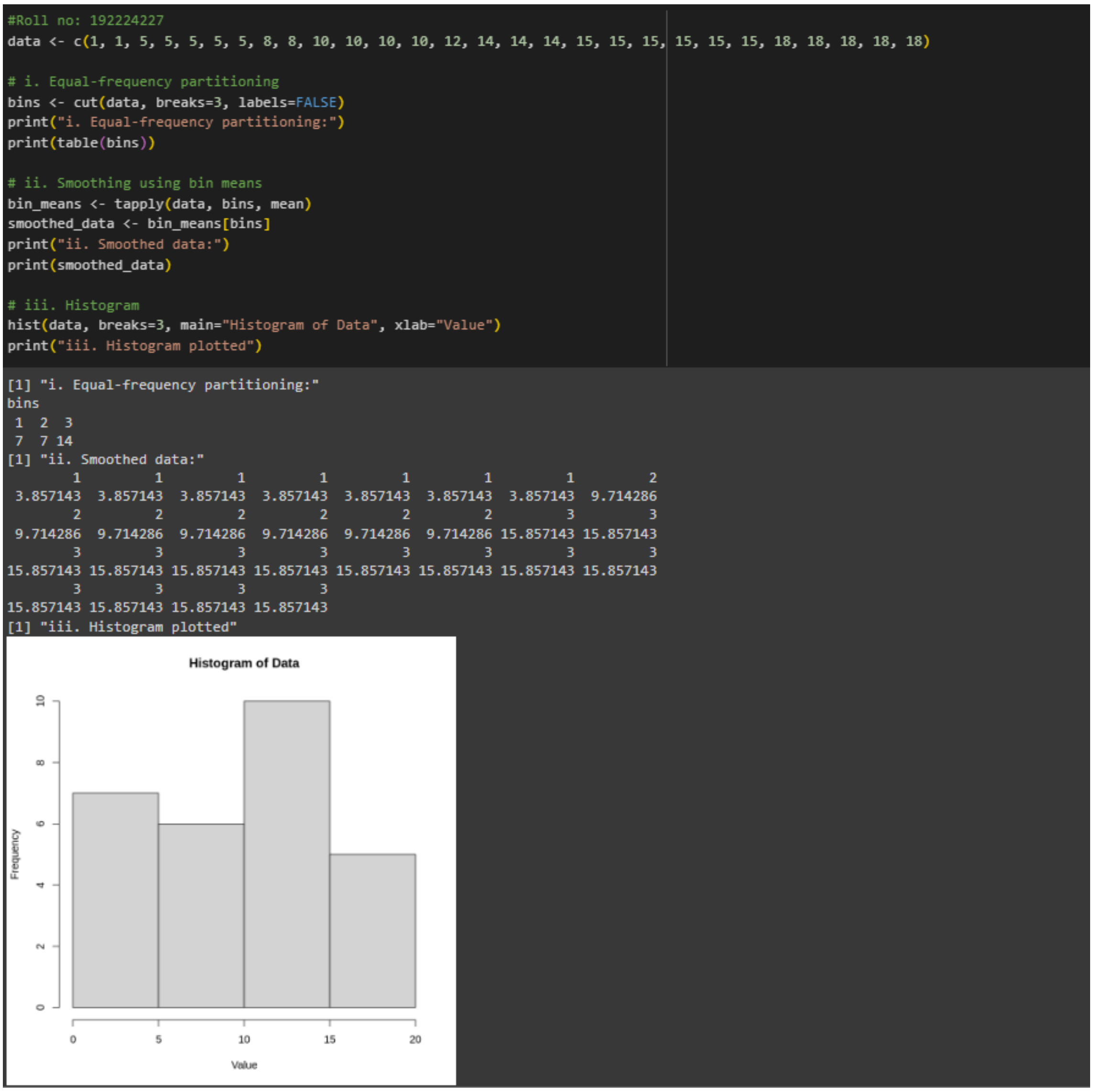
Question-1



```
#Reg No: 192224227
preferences <- matrix(c(18, 22, 20, 2, 28, 40, 20, 10, 40), nrow=3, byrow=TRUE)
colnames(preferences) <- c("A", "B", "C")

# 1. Covariance between B and C
cov_bc <- cov(preferences[,"B"], preferences[,"C"])
print("1. Covariance between B and C:")
print(cov_bc)

# 2. Covariance matrix
cov_matrix <- cov(preferences)
print("2. Covariance matrix:")
print(cov_matrix)

# 3. Correlation between B and C
cor_bc <- cor(preferences[,"B"], preferences[,"C"])
print("3. Correlation between B and C:")
print(cor_bc)

# 4. Correlation matrix
cor_matrix <- cor(preferences)
print("4. Correlation matrix:")
print(cor_matrix)
```

```
[1] "1. Covariance between B and C:"
[1] -20
[1] "2. Covariance matrix:"
          A   B         C
A  97.33333 -74 -46.66667
B -74.00000  84 -20.00000
C -46.66667 -20 133.33333
[1] "3. Correlation between B and C:"
[1] -0.1889822
[1] "4. Correlation 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
```

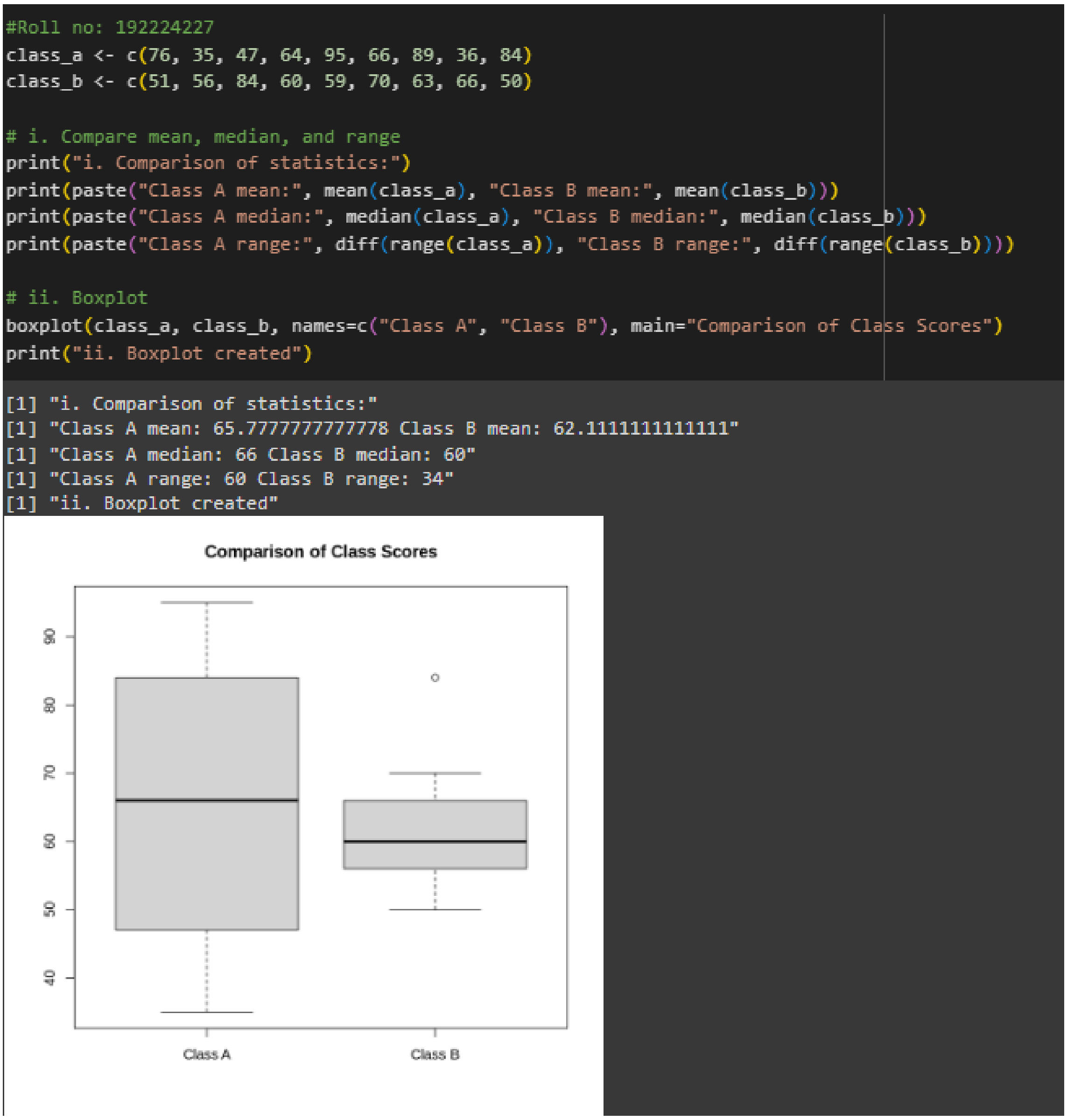
Question-2



```
#Roll no: 192224227
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)

# i. Equal-frequency partitioning
bins <- cut(data, breaks=3, labels=FALSE)
print("i. Equal-frequency partitioning:")
print(table(bins))

# ii. Smoothing using bin means
bin_means <- tapply(data, bins, mean)
smoothed_data <- bin_means[bins]
print("ii. Smoothed data:")
print(smoothed_data)

# iii. Histogram
hist(data, breaks=3, main="Histogram of Data", xlab="Value")
print("iii. Histogram plotted")
```

```
[1] "i. Equal-frequency partitioning:"
bins
 1  2  3 
 7  7 14 
[1] "ii. Smoothed data:"
        1         1         1         1         1         1         1         2 
 3.857143  3.857143  3.857143  3.857143  3.857143  3.857143  3.857143  9.714286 
        2         2         2         2         2         2         3         3 
 9.714286  9.714286  9.714286  9.714286  9.714286  9.714286 15.857143 15.857143 
        3         3         3         3         3         3         3         3 
15.857143 15.857143 15.857143 15.857143 15.857143 15.857143 15.857143 15.857143 
        3         3         3         3 
15.857143 15.857143 15.857143 15.857143 
[1] "iii. Histogram plotted"
```

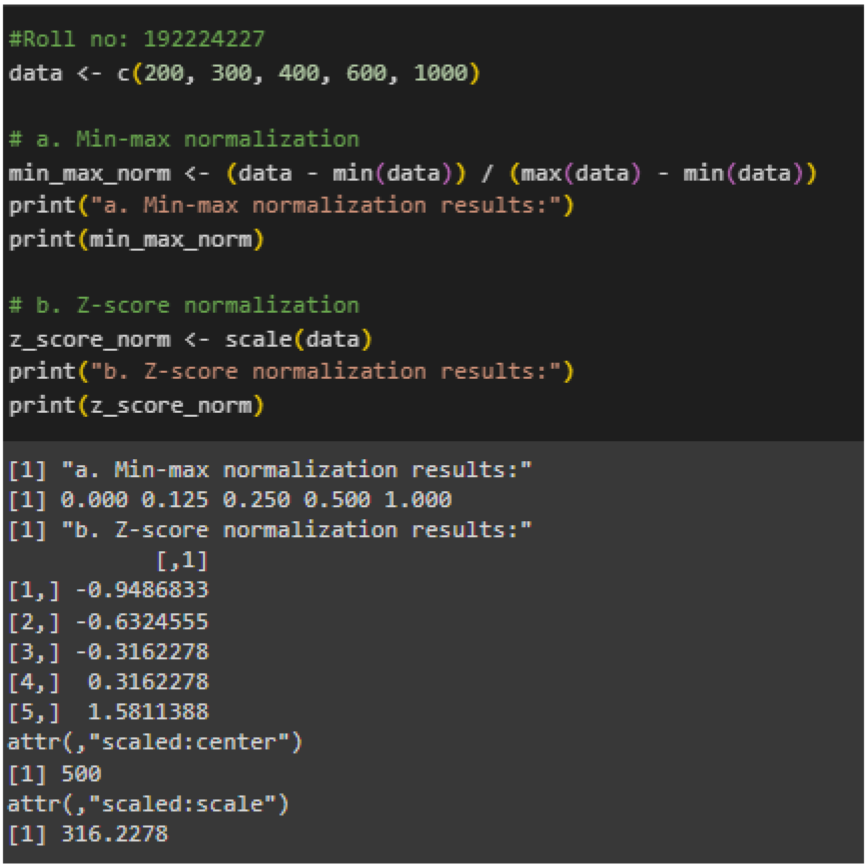
Question-3



```
#Roll no: 192224227
class_a <- c(76, 35, 47, 64, 95, 66, 89, 36, 84)
class_b <- c(51, 56, 84, 60, 59, 70, 63, 66, 50)

# i. Compare mean, median, and range
print("i. Comparison of statistics:")
print(paste("Class A mean:", mean(class_a), "Class B mean:", mean(class_b)))
print(paste("Class A median:", median(class_a), "Class B median:", median(class_b)))
print(paste("Class A range:", diff(range(class_a)), "Class B range:", diff(range(class_b))))

# ii. Boxplot
boxplot(class_a, class_b, names=c("Class A", "Class B"), main="Comparison of Class Scores")
print("ii. Boxplot created")
```

```
[1] "i. Comparison of statistics:"
[1] "Class A mean: 65.7777777777778 Class B mean: 62.1111111111111"
[1] "Class A median: 66 Class B median: 60"
[1] "Class A range: 60 Class B range: 34"
[1] "ii. Boxplot created"
```

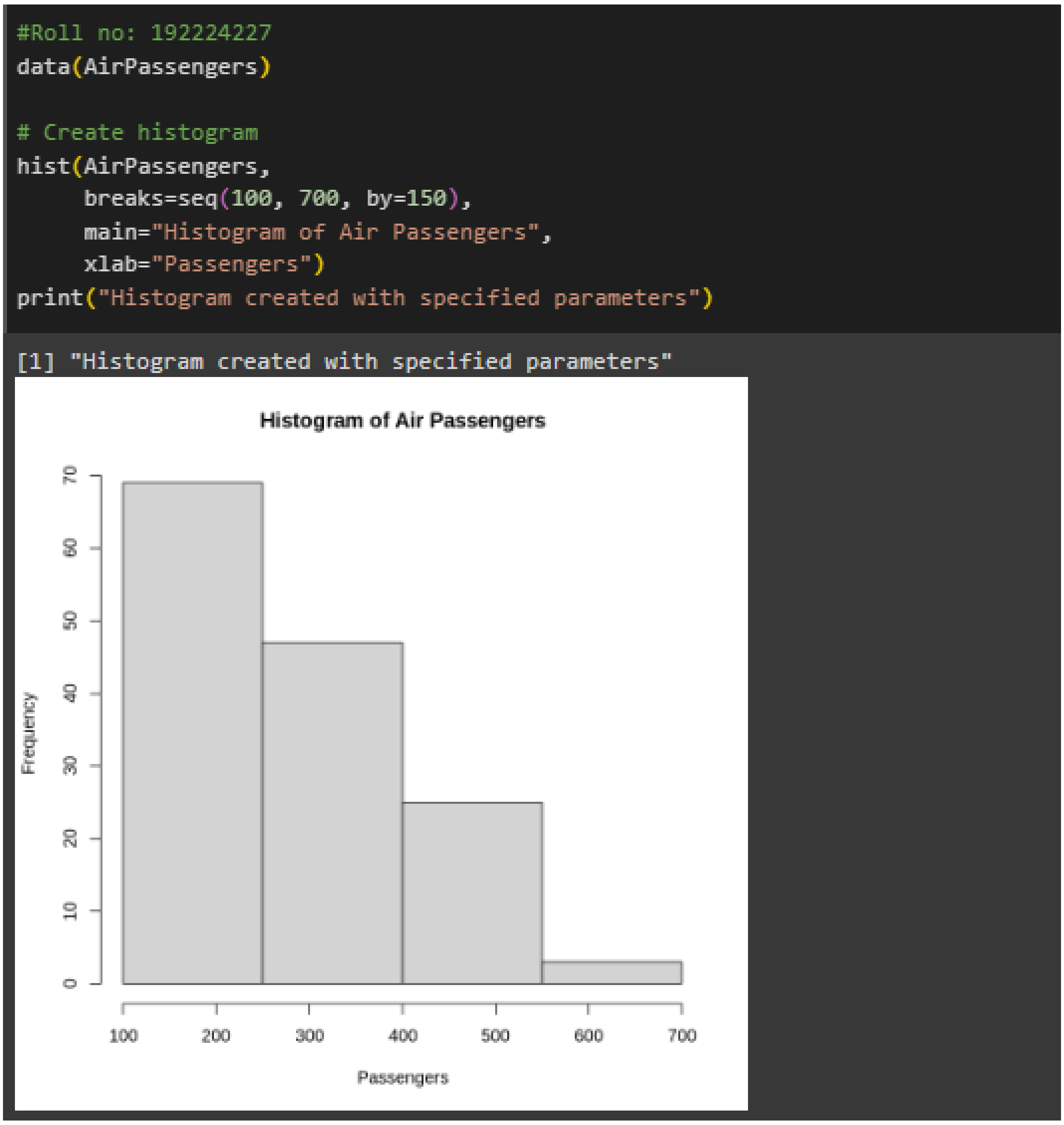
Question-4



```
#Roll no: 192224227
data <- c(200, 300, 400, 600, 1000)

# a. Min-max normalization
min_max_norm <- (data - min(data)) / (max(data) - min(data))
print("a. Min-max normalization results:")
print(min_max_norm)

# b. Z-score normalization
z_score_norm <- scale(data)
print("b. Z-score normalization results:")
print(z_score_norm)
```

```
[1] "a. Min-max normalization results:"
[1] 0.000 0.125 0.250 0.500 1.000
[1] "b. Z-score normalization results:"
           [,1]
[1,] -0.9486833
[2,] -0.6324555
[3,] -0.3162278
[4,]  0.3162278
[5,]  1.5811388
attr(,"scaled:center")
[1] 500
attr(,"scaled:scale")
[1] 316.2278
```

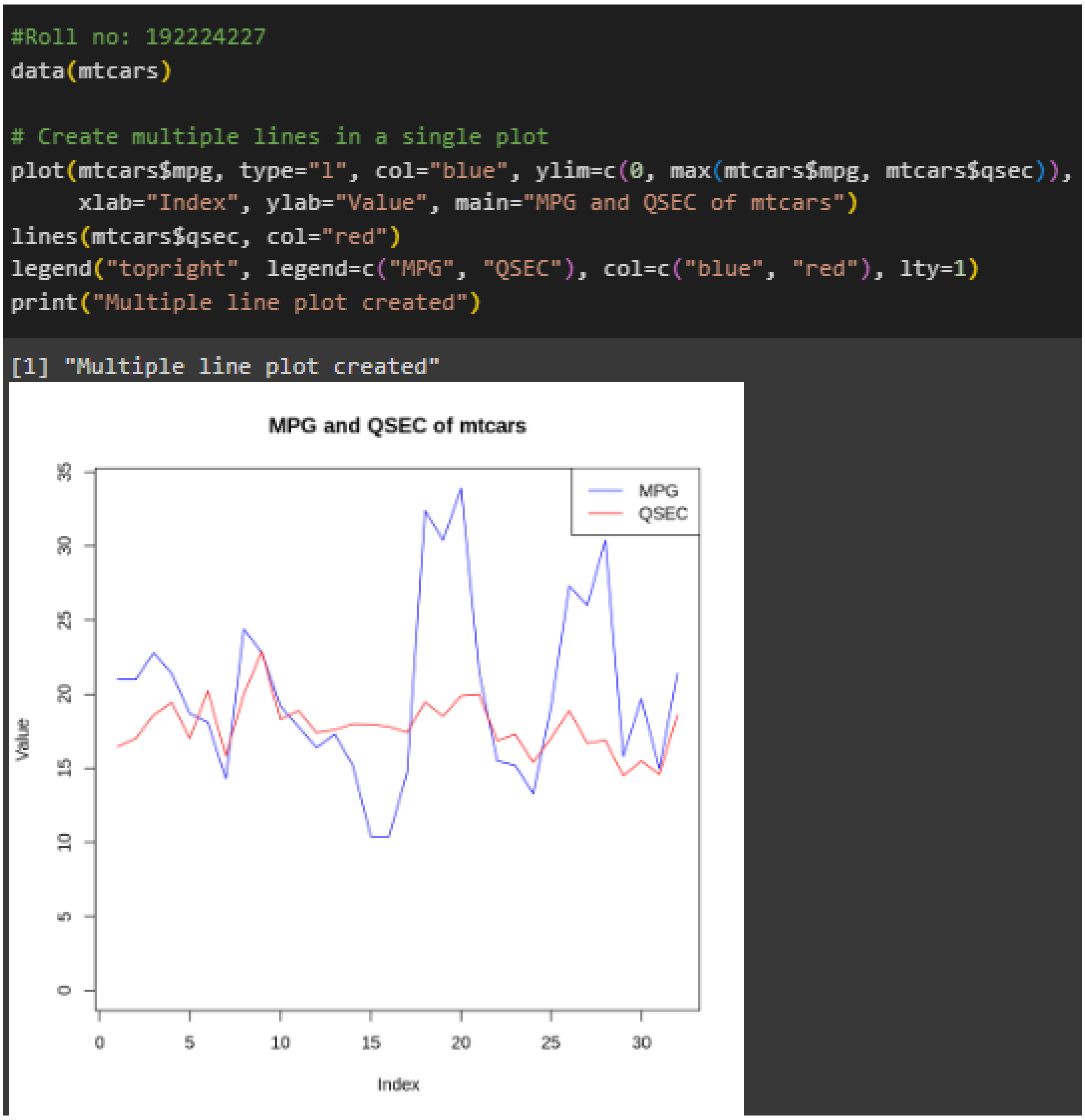
Question-5



```
#Roll no: 192224227
data(AirPassengers)

# Create histogram
hist(AirPassengers,
     breaks=seq(100, 700, by=150),
     main="Histogram of Air Passengers",
     xlab="Passengers")
print("Histogram created with specified parameters")
```

```
[1] "Histogram created with specified parameters"
```

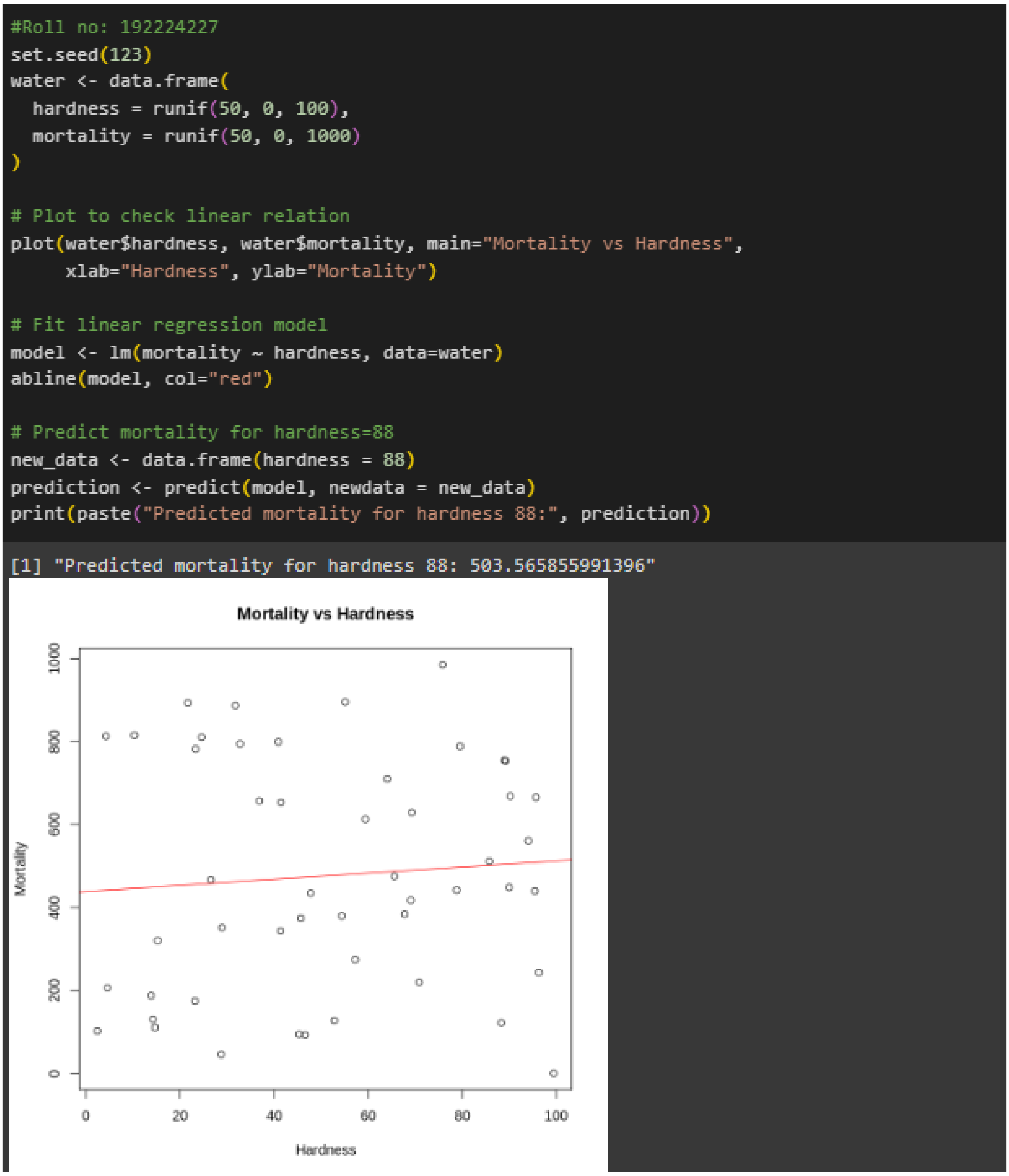
Question-6



```
#Roll no: 192224227
data(mtcars)

# Create multiple lines in a single plot
plot(mtcars$mpg, type="l", col="blue", ylim=c(0, max(mtcars$mpg, mtcars$qsec)),
     xlab="Index", ylab="Value", main="MPG and QSEC of mtcars")
lines(mtcars$qsec, col="red")
legend("topright", legend=c("MPG", "QSEC"), col=c("blue", "red"), lty=1)
print("Multiple line plot created")
```

```
[1] "Multiple line plot created"
```

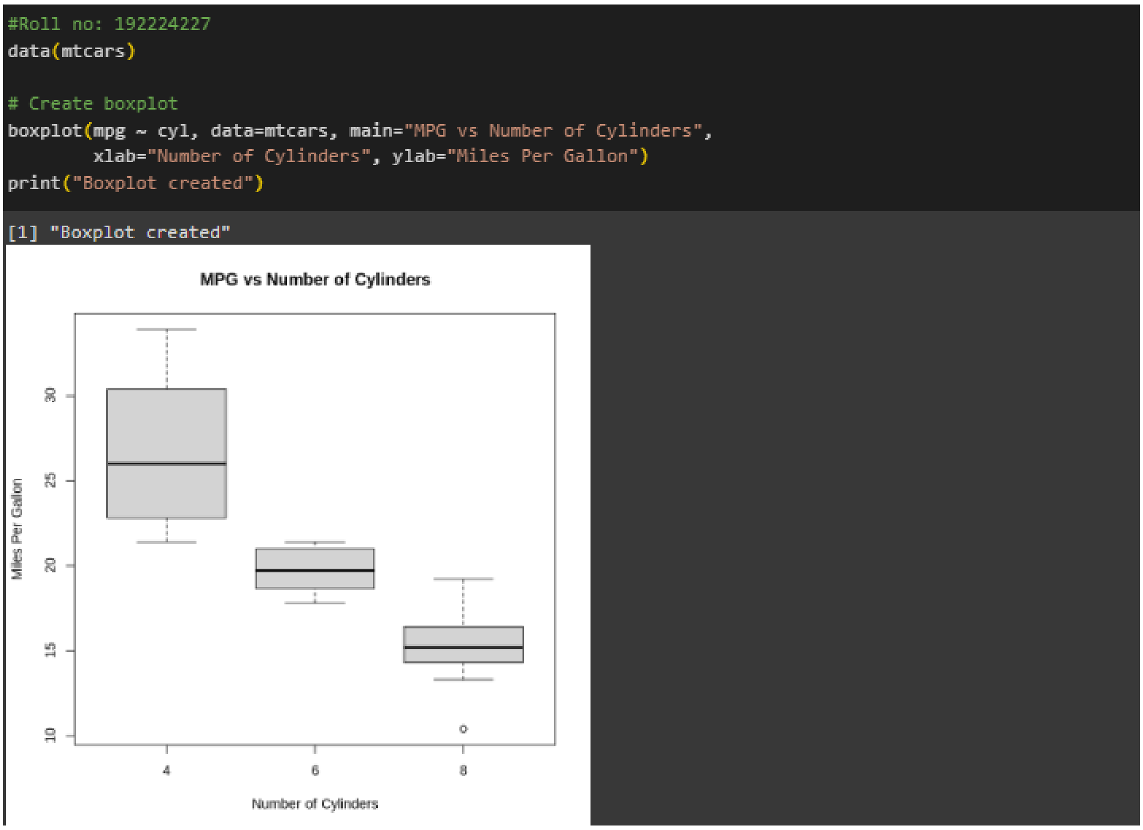
Question-7



```
#Roll no: 192224227
set.seed(123)
water <- data.frame(
  hardness = runif(50, 0, 100),
  mortality = runif(50, 0, 1000)
)

# Plot to check linear relation
plot(water$hardness, water$mortality, main="Mortality vs Hardness",
     xlab="Hardness", ylab="Mortality")

# Fit linear regression model
model <- lm(mortality ~ hardness, data=water)
abline(model, col="red")

# Predict mortality for hardness=88
new_data <- data.frame(hardness = 88)
prediction <- predict(model, newdata = new_data)
print(paste("Predicted mortality for hardness 88:", prediction))
```

```
[1] "Predicted mortality for hardness 88: 503.565855991396"
```

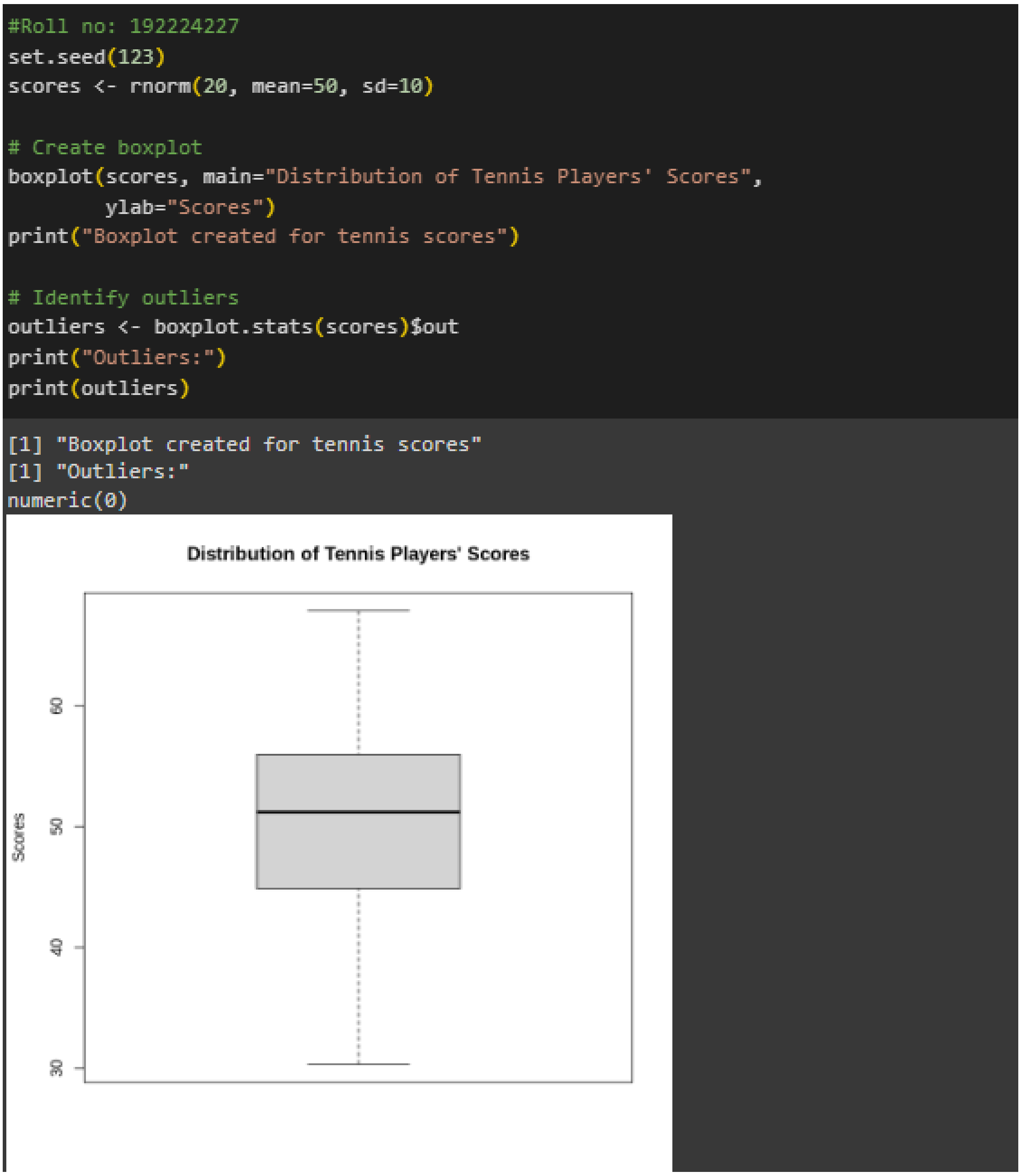
Question-8



```
#Roll no: 192224227
data(mtcars)

# Create boxplot
boxplot(mpg ~ cyl, data=mtcars, main="MPG vs Number of Cylinders",
        xlab="Number of Cylinders", ylab="Miles Per Gallon")
print("Boxplot created")
```

```
[1] "Boxplot created"
```

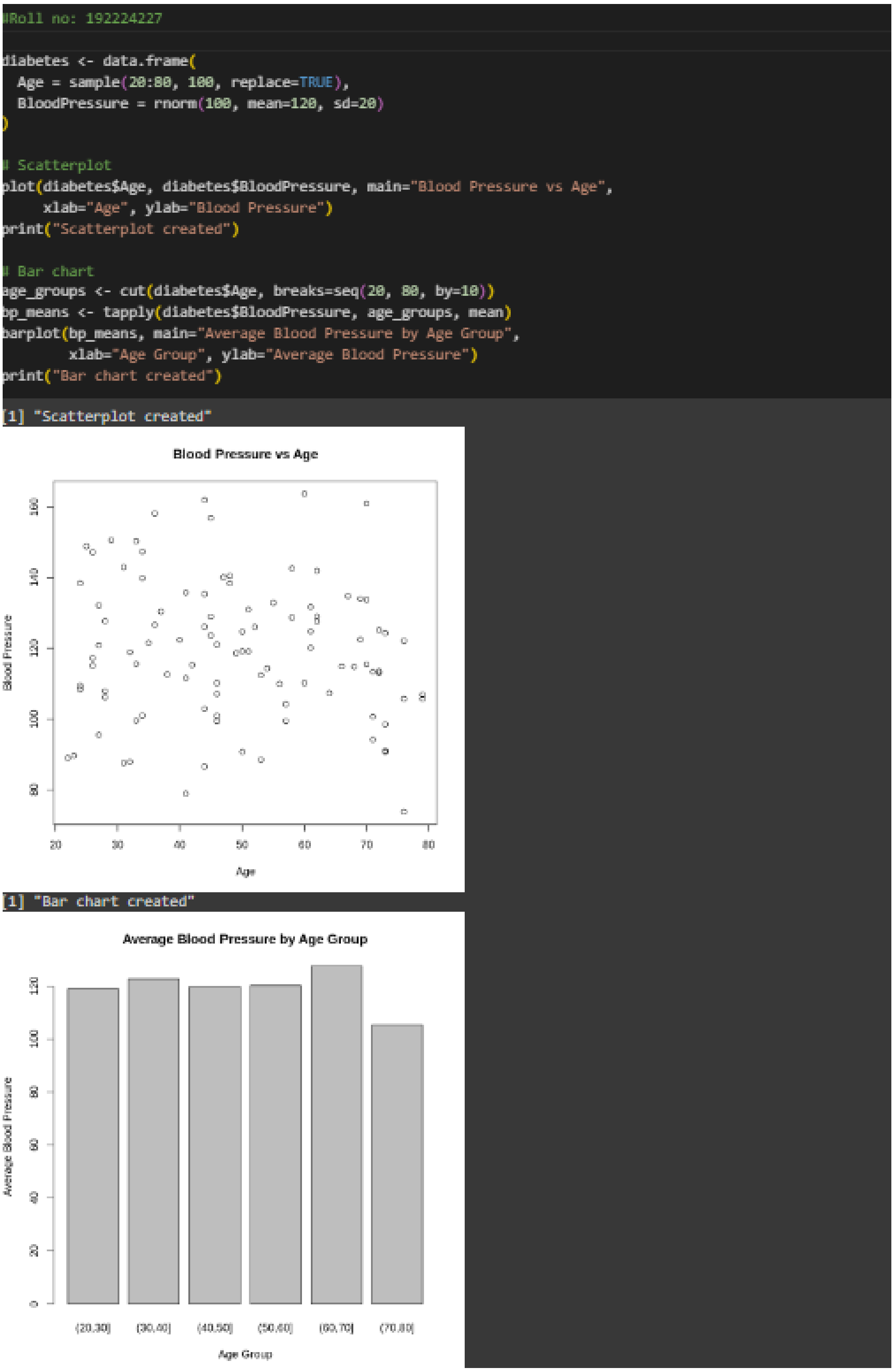
Question-9



```
#Roll no: 192224227
set.seed(123)
scores <- rnorm(20, mean=50, sd=10)

# Create boxplot
boxplot(scores, main="Distribution of Tennis Players' Scores",
        ylab="Scores")
print("Boxplot created for tennis scores")

# Identify outliers
outliers <- boxplot.stats(scores)$out
print("Outliers:")
print(outliers)
```

```
[1] "Boxplot created for tennis scores"
[1] "Outliers:"
numeric(0)
```

Question-10



```
#Roll no: 192224227

diabetes <- data.frame(
  Age = sample(20:80, 100, replace=TRUE),
  BloodPressure = rnorm(100, mean=120, sd=20)
)

# Scatterplot
plot(diabetes$Age, diabetes$BloodPressure, main="Blood Pressure vs Age",
     xlab="Age", ylab="Blood Pressure")
print("Scatterplot created")

# Bar chart
age_groups <- cut(diabetes$Age, breaks=seq(20, 80, by=10))
bp_means <- tapply(diabetes$BloodPressure, age_groups, mean)
barplot(bp_means, main="Average Blood Pressure by Age Group",
        xlab="Age Group", ylab="Average Blood Pressure")
print("Bar chart created")
```

```
[1] "Scatterplot created"
```

```
[1] "Bar chart created"
```