BT307 Biological Data Analysis Lab Assignment No. 3 08.02.2024

Note: Submit the assignment online through Moodle either in .doc or .pdf format. Your final report file should be named as "**YourName_BT307_Lab3_08022024**". Make sure that your name and roll numbers are written at the first page of your final report. Note that you can upload only one file; thus, put together all the answers in a single file.

Goal of this exercise is to learn about the basic plots in R.

(1) # Import the data using read.csv()

```
SPKData = read.csv("CardioGoodFitness.csv", stringsAsFactors = F)
```

##

(2) # Create a scatter plot between Age and Income.

```
plot(SPKData$Age, SPKData$Income)
```

(3) # Modify the plot with labels and so on

```
plot(SPKData$Age, SPKData$Income,
    main = "Ags Vs Income", xlab = "Age in years", ylab = "Income in dollars",
    pch = 20, col = "red", cex = 1, cex.axis = 1.0, cex.lab = 1.0)
```

(4) # Since there are multiple values of income for each age, let us take average

```
income_avg <- aggregate(Income ~ Age, SPKData, mean)
```

(5) # Now, plot the Age vs Income

```
plot(income_avg$Age, income_avg$Income,
    main = "Ags Vs Income", xlab = "Age in years", ylab = "Income in dollars",
    pch = 20, col = "red", cex = 2, cex.axis = 1.0, cex.lab = 1.0)
```

(6) # We can also adjust the axes scales

```
plot(income_avg$Age, income_avg$Income,
    main = "Ags Vs Income", xlab = "Age in years", ylab = "Income in dollars",
    pch = 20, col = "red", cex = 2, cex.axis = 1.0, cex.lab = 1.0,
    xlim = c(15, 50), ylim = c(25000, 90000))
```

(7) # Let us plot the same data as line graph

```
plot(income_avg$Age, income_avg$Income,
    main = "Ags Vs Income", xlab = "Age in years", ylab = "Income in dollars",
    pch = 20, col = "red", cex = 2, cex.axis = 1.0, cex.lab = 1.0,
    xlim = c(15, 50), ylim = c(25000, 90000), type = "b", lwd = 2)
```

##

(8) # Now, let us try to plot the bar graph.

```
product <- read.csv("Product_Avg.csv", stringsAsFactors = F)
```

(9) # First let us make a bar plot for product TM195

```
product.tm195 <- product[1, 2:7]
product.tm195 <- data.matrix(product.tm195)
barplot(product.tm195,
      ylim = c(0, 50000), ylab = "Average values")
```

(10) # Due to the large difference between the income and other variables, it is not clear. So, let us remove income for the time-being.

```
product.tm195 <- product[1, 2:6]
product.tm195 <- data.matrix(product.tm195)
barplot(product.tm195,
      ylim = c(0, 100), ylab = "Average values")
```

(11) # Now let us include all the three product types

```
product.all <- data.matrix(product[ , 2:6])
barplot(product.all,
      beside = TRUE, ylim = c(0, 200), ylab = "Average values",
      legend.text = c("TM195", "TM498", "TM798"),
      args.legend = list(bty = "n", x = "topleft"))
```

(12) # Let us add some more arguments

```
mycol <- c("red", "blue", "green")
barplot(product.all,
      beside = TRUE, ylim = c(0, 200), ylab = "Average values",
      legend.text = c("TM195", "TM498", "TM798"),
      args.legend = list(bty = "n", x = "topleft"),
      col = mycol, cex.axis = 1.25, cex.name = 1.25, cex.lab = 1.25)
```

##

(13) # Now let us try histogram plot

```
hist(SPKData$Income)
```

(14) # Now we can do some formatting and plot

```
hist(SPKData$Income,
    xlim = c(20000, 120000), ylim = c(0,60), xlab = "Income in dollar",
    ylab = "Frequency", main = "Frequency distribution of Income", las = 1)
```

(15) # By default, the no. of bins is calculated using Sturges's method. This time, let us use another rule called Freedman-Diaconis rule.

```
hist(SPKData$Income,
    xlim = c(20000, 120000), ylim = c(0,60), xlab = "Income in dollar", ylab = "Frequency",
main = "Frequency distribution of Income - Freeman-Diaconis method", las = 1,
    breaks = "FD")
```

(16) # We can use Scott method as well.

```
hist(SPKData$Income,
    xlim = c(20000, 120000), ylim = c(0,60), xlab = "Income in dollar", ylab = "Frequency",
    main = "Frequency distribution of Income - Scott method", las = 1, breaks = "Scott")
```

(17) # In a frequency distribution, we can mark mean and median

```
abline(v = mean(SPKData$Income), lwd = 2, col = "red")
abline(v = median(SPKData$Income), lwd = 2, col = "blue")
```

(18) # Now, let us change the y-axis (frequency or count) to density.

```
hist(SPKData$Income,
    xlim = c(20000, 120000), xlab = "Income in dollar", ylab = "Density",
    main = "Frequency distribution of Income - Freeman-Diaconis method",
```

```
    las = 1, breaks = "FD", freq = FALSE)
```

(19) # Now, let us change (frequency or count) to percentage value in y-axis.

```
percentage <- hist(SPKData$Income, plot = FALSE)

# convert density to percentage
percentage$density <- percentage$counts*100/sum(percentage$counts)
```

(20) # Now plot the histogram with percentage

```
plot(percentage,
    xlim = c(20000, 120000), ylim = c(0, 60), xlab = "Income in dollar",
    ylab = "Income (%)",
    main = "Frequency distribution of Income - Freeman-Diaconis method",
    las = 1, col = "yellow")
```

```
##########################################################################
```

(21) # Now let us try to plot the box plot

```
bp <- SPKData
boxplot(Income ~ Product, bp)
```

(22) # We can do some formatting in the box plot

```
boxplot(Income ~ Product, bp,
      xlab = "Product type", ylab = "Income", cex.lab = 1.0, cex.axis = 1.0,
      ylim = c(20000, 110000), varwidth = TRUE)
```

(23) # Side-by-side box plot

```
age <- SPKData$Age
education <- SPKData$Education
rmiles <- SPKData$Miles
miles <- rmiles/5
age_education_miles = cbind(age, education, miles)
boxplot(age_education_miles, beside=T)
```

(24) # We can format little bit

```
boxplot(age_education_miles, beside=T,
      ylim = c(0, 80), varwidth = TRUE, outpch = 19, outcol = "red")
```

```
##########################################################################
```

(25) # Now let us try to plot heat map

```
hm <- data.matrix(SPKData)
heatmap(hm[ , 2:7], Rowv = NA, Colv = NA)
```

(26) # Generate the dendogram

```
heatmap(hm[ , 2:7])
```

(27) # We can scale the data

```
heatmap(hm[ , 2:7], scale = "column")
```

(28) # We can remove the horizontal dendogram

```
heatmap(hm[ , 2:7], Rowv = NA, scale = "column")
```