RWorksheets_Bagilidad#4b

2023-11-08

#1. Using the for loop, create an R script that will display a 5x5 matrix as shown in # Figure 1. It must contain vectorA = [1,2,3,4,5] and a 5 x 5 zero matrix.

```
vectorA <- c(1,2,3,4,5)

matrixA <- matrix(0,nrow = 5, ncol =5)

for (i in 1:5)
  for (j in 1:5)
  {
    matrixA[i,j] <- abs (vectorA[i] - vectorA[j])
  }

matrixA
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    2    3    4
## [2,]    1    0    1    2    3
## [3,]    2    1    0    1    2
## [4,]    3    2    1    0    1
## [5,]    4    3    2    1    0
```

2. Print the string "*" using for() function. The output should be the same as shown

in Figure 2.

```
for (i in 1:5) {
  cat(paste0("\"", rep("*", i), "\""), "\n")
}
```

```
## "*"
## "*" "*"
## "*" "*" "*"
## "*" "*" "*" "*"
## "*" "*" "*" "*" "*"
```

#3. Get an input from the user to print the Fibonacci sequence starting from the 1st input # up to 500. Use repeat and break statements. Write the R Scripts and its output.

```
begin <- as.numeric(readline("Enter the starting number for the Fibonacci sequence: "))
```

```
## Enter the starting number for the Fibonacci sequence:
```

```
a <- 0
b <- 1
```

```r
cat("Fibonacci sequence starting from", begin, ":\n")
```

```
## Fibonacci sequence starting from NA :
```

```r
cat(begin, "")
```

```
## NA
```

```r
repeat {
  nextFib <- a + b
  if (nextFib > 500) {
    break
  }
  cat(nextFib, " ")
  a <- b
  b <- nextFib
}
```

```
## 1  2  3  5  8  13  21  34  55  89  144  233  377
```

```r
cat("\n")
```

#4. #a. What is the R script for importing an excel or a csv file? Display the first 6 rows of the dataset? Show your codes and its result

```r
data_shoes <- read.csv("data_shoes.csv", header = TRUE)
data_shoes
```

```
##     X Shoe_Size Height Gender
## 1   1       6.5   66.0      F
## 2   2       9.0   68.0      F
## 3   3       8.5   64.5      F
## 4   4       8.5   65.0      F
## 5   5      10.5   70.0      M
## 6   6       7.0   64.0      F
## 7   7       9.5   70.0      F
## 8   8       9.0   71.0      F
## 9   9      13.0   72.0      M
## 10 10       7.5   64.0      F
## 11 11      10.5   74.5      M
## 12 12       8.5   67.0      F
## 13 13      12.0   71.0      M
## 14 14      10.5   71.0      M
## 15 15      13.0   77.0      M
## 16 16      11.5   72.0      M
## 17 17       8.5   59.0      F
## 18 18       5.0   62.0      F
## 19 19      10.0   72.0      M
## 20 20       6.5   66.0      F
## 21 21       7.5   64.0      F
## 22 22       8.5   67.0      M
## 23 23      10.5   73.0      M
## 24 24       8.5   69.0      F
## 25 25      10.5   72.0      M
## 26 26      11.0   70.0      M
## 27 27       9.0   69.0      M
```

```
## 28 28      13.0   70.0      M
```

b.Create a subset for gender(female and male). How many observations are there in Male? How about in Female? Write the R scripts and its output.

```r
data_shoes <- read.csv("data_shoes.csv")
data_shoes
```

```
##     X Shoe_Size Height Gender
## 1   1       6.5   66.0      F
## 2   2       9.0   68.0      F
## 3   3       8.5   64.5      F
## 4   4       8.5   65.0      F
## 5   5      10.5   70.0      M
## 6   6       7.0   64.0      F
## 7   7       9.5   70.0      F
## 8   8       9.0   71.0      F
## 9   9      13.0   72.0      M
## 10 10       7.5   64.0      F
## 11 11      10.5   74.5      M
## 12 12       8.5   67.0      F
## 13 13      12.0   71.0      M
## 14 14      10.5   71.0      M
## 15 15      13.0   77.0      M
## 16 16      11.5   72.0      M
## 17 17       8.5   59.0      F
## 18 18       5.0   62.0      F
## 19 19      10.0   72.0      M
## 20 20       6.5   66.0      F
## 21 21       7.5   64.0      F
## 22 22       8.5   67.0      M
## 23 23      10.5   73.0      M
## 24 24       8.5   69.0      F
## 25 25      10.5   72.0      M
## 26 26      11.0   70.0      M
## 27 27       9.0   69.0      M
## 28 28      13.0   70.0      M
```

```r
male_subset <- subset(data_shoes,Gender=="M")
male_subset
```

```
##     X Shoe_Size Height Gender
## 5   5      10.5   70.0      M
## 9   9      13.0   72.0      M
## 11 11      10.5   74.5      M
## 13 13      12.0   71.0      M
## 14 14      10.5   71.0      M
## 15 15      13.0   77.0      M
## 16 16      11.5   72.0      M
## 19 19      10.0   72.0      M
## 22 22       8.5   67.0      M
## 23 23      10.5   73.0      M
```

```
## 25 25      10.5   72.0      M
## 26 26      11.0   70.0      M
## 27 27       9.0   69.0      M
## 28 28      13.0   70.0      M
```

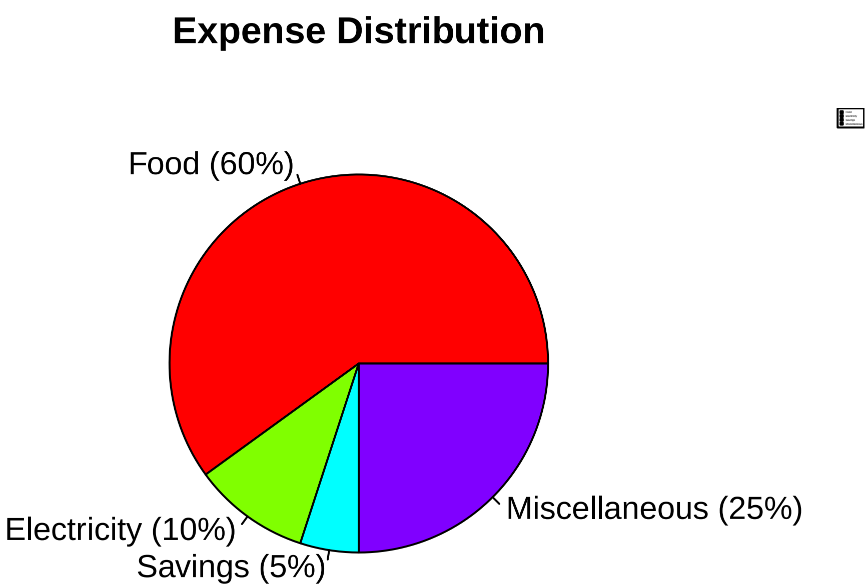
```r
female_subset <- subset(data_shoes,Gender=="F")
female_subset
```

```
##     X Shoe_Size Height Gender
## 1   1       6.5   66.0      F
## 2   2       9.0   68.0      F
## 3   3       8.5   64.5      F
## 4   4       8.5   65.0      F
## 6   6       7.0   64.0      F
## 7   7       9.5   70.0      F
## 8   8       9.0   71.0      F
## 10 10       7.5   64.0      F
## 12 12       8.5   67.0      F
## 17 17       8.5   59.0      F
## 18 18       5.0   62.0      F
## 20 20       6.5   66.0      F
## 21 21       7.5   64.0      F
## 24 24       8.5   69.0      F
```

#5.

#a. Create a piechart that will include labels in percentage.Add some colors and title of chart. Write the R scripts and show its output.

```r
# Define the data
data <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)

# Calculate percentages and format them as strings
percentages <- paste(round(100 * data / sum(data), 1), "%", sep = "")

# Create a pie chart
pie(data,
    labels = paste(names(data),
    " (", percentages, ")", sep = ""),
    col = rainbow(length(data)),
    main = "Expense Distribution")

# Add a legend
legend("topright",
       names(data),
       cex = 0.10,
       fill = rainbow(length(data)))
```



#6. #a. Check for the structure of the data set using the str() function. Describe what you have seen in the output.

```
data(iris)
str(iris)
```

```
## 'data.frame':	150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

#b.

```
mean_Iris <- c(mean(iris$Sepal.Length), mean(iris$Sepal.Width), mean(iris$Petal.Length), mean(iris$Petal
mean_Iris
```

```
## [1] 5.843333 3.057333 3.758000 1.199333
```

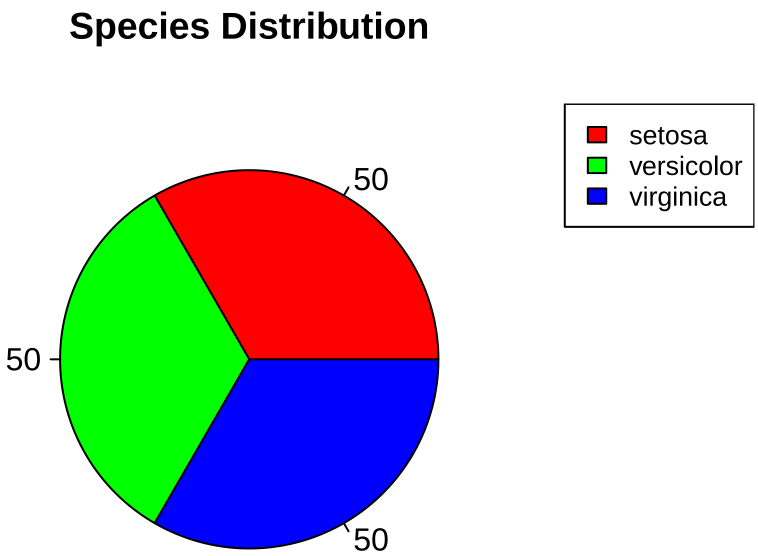
```
mean_sepal_length <- mean(iris$Sepal.Length)
mean_sepal_width <- mean(iris$Sepal.Width)
mean_petal_length <- mean(iris$Petal.Length)
mean_petal_width <- mean(iris$Petal.Width)

mean_Iris <- data.frame(Sepal_Length = mean_sepal_length,
                        Sepal_Width = mean_sepal_width,
                        Petal_Length = mean_petal_length,
                        Petal_Width = mean_petal_width)
mean_Iris
```

```
##   Sepal_Length Sepal_Width Petal_Length Petal_Width
## 1     5.843333    3.057333        3.758    1.199333
```

#c. Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script and its result.

```
species_count <- table(iris$Species)
pie(species_count, labels = species_count, col = rainbow(length(species_count)), main = "Species Distri
```

```r
legend("topright", names(species_count), cex = 0.8, fill = rainbow(length(species_count)))
```



#d.

```r
# Subset the iris data set into the three species.
setosa_subset <- subset(iris, Species == "setosa")
versicolor_subset <- subset(iris, Species == "versicolor")
virginica_subset <- subset(iris, Species == "virginica")

# Display the last six rows of each species.
tail(setosa_subset, 6)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 45          5.1         3.8          1.9         0.4  setosa
## 46          4.8         3.0          1.4         0.3  setosa
## 47          5.1         3.8          1.6         0.2  setosa
## 48          4.6         3.2          1.4         0.2  setosa
## 49          5.3         3.7          1.5         0.2  setosa
## 50          5.0         3.3          1.4         0.2  setosa
```

```r
tail(versicolor_subset, 6)
```

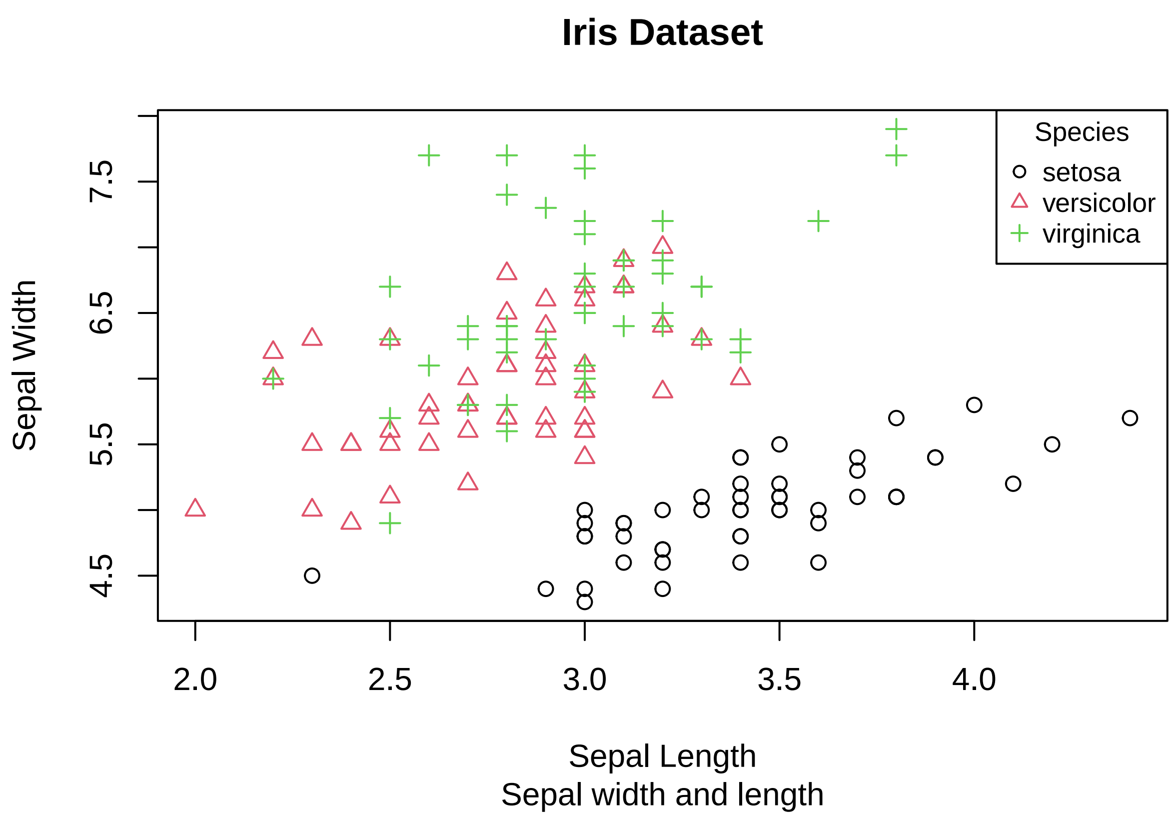
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         2.9          4.3         1.3 versicolor
## 99           5.1         2.5          3.0         1.1 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
```

```r
tail(virginica_subset, 6)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

#e. Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versicolor,virginica). Add a title = "Iris Dataset", subtitle = "Sepal width and length, labels for the x and y axis, the pch symbol and colors should be based on the species.

```
# Convert the "Species" column to a factor
iris$Species <- as.factor(iris$Species)

# Create a scatterplot
plot(
  Sepal.Length ~ Sepal.Width,
  data = iris,
  pch = as.integer(iris$Species),  # Use different pch symbols for each species
  col = as.integer(iris$Species),  # Use different colors for each species
  xlab = "Sepal Length",
  ylab = "Sepal Width",
  main = "Iris Dataset",
  sub = "Sepal width and length"
)

# Add a legend
legend("topright", legend = levels(iris$Species), col = 1:3, pch = 1:3, cex = 0.8, title = "Species")
```



#f.Interpret the result. # The dataset consists of five variables (columns) and 150 observations (rows) in a data frame format. # Petal.Length, Petal.Width, Sepal.Length, and Sepal. Width are the names of the four numerical variables. These stand for the measures of the length and width of the petals, respectively. # The factor variable Species, which represents the species of iris flowers, is the sixth variable. There are three tiers to it: "setosa," "versicolor," and "virginica," denoting the many iris flower species included in the dataset.

#7.Import the alexa-file.xlsx. Check on the variations. Notice that there are extra whitespaces among black variants (Black Dot, Black Plus, Show, Black Spot). Also on the white variants (White Dot, White

Plus, White Show, White Spot).