RWorksheet_Lego#4b

2023-11-08

##Using Loop Function for() loop ##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(1:5)
matrix<- matrix(0,5,5)
vectorA
```

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

```
matrix
```

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

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

```
##      [,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

```
size<- 5
for (i in 1:size){
  cat(paste(rep("*",i), collapse=""), "\n")
}
```

```
## *
## **
## ***
## ****
## *****
```

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

```
fib<- function(starting_value){
  a<-0
  b<-1

  repeat{
    value<- a+b

  if( value > 500){
    break
  }
    cat(value,"")

    a<-b
    b<-value
  }
}
user_input<-as.numeric(readline(prompt= "Enter the starting value for Fibonacci sequence: "))
```

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

```
result<-fib(user_input)
```

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

```
fibonacci <- function(n) {
  a <- 0
  b <- 1
  fib_seq <- c(a, b)

  repeat {
    next_term <- a + b
    if (next_term > 500) {
      break
    }
    fib_seq <- c(fib_seq, next_term)
    a <- b
    b <- next_term
  }

  return(fib_seq)
}

user_input <- as.numeric(readline(prompt = "Enter the starting point for Fibonacci sequence: "))
```

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

```
result <- fibonacci(user_input)
print(result)
```

```
##  [1]   0   1   1   2   3   5   8  13  21  34  55  89 144 233 377
```

##4. Import the dataset as shown in Figure 1 you have created previously. ##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

```
library(readr)
Shoe_Data <- read_csv("Shoe Data.csv")
```

```
## Rows: 28 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Gender
## dbl (2): Shoe Size, Height
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
Shoe_Data
```

```
## # A tibble: 28 x 3
##    `Shoe Size` Height Gender
##          <dbl>  <dbl> <chr>
##  1         6.5   66   F
##  2         9     68   F
##  3         8.5   64.5 F
##  4         8.5   65   F
##  5        10.5   70   M
##  6         7     64   F
##  7         9.5   70   F
##  8         9     71   F
##  9        13     72   M
## 10         7.5   64   F
## # i 18 more rows
```

```r
head(Shoe_Data, n=6)
```

```
## # A tibble: 6 x 3
##   `Shoe Size` Height Gender
##         <dbl>  <dbl> <chr>
## 1         6.5   66   F
## 2         9     68   F
## 3         8.5   64.5 F
## 4         8.5   65   F
## 5        10.5   70   M
## 6         7     64   F
```

##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
subsetMale<- subset(Shoe_Data, Gender == "M")
subsetMale
```

```
## # A tibble: 14 x 3
##    `Shoe Size` Height Gender
##          <dbl>  <dbl> <chr>
##  1        10.5   70   M
##  2        13     72   M
##  3        10.5   74.5 M
##  4        12     71   M
##  5        10.5   71   M
##  6        13     77   M
##  7        11.5   72   M
##  8        10     72   M
##  9         8.5   67   M
## 10        10.5   73   M
## 11        10.5   72   M
## 12        11     70   M
## 13         9     69   M
## 14        13     70   M
```

```
subsetFemale<- subset(Shoe_Data, Gender == "F")
subsetFemale
```

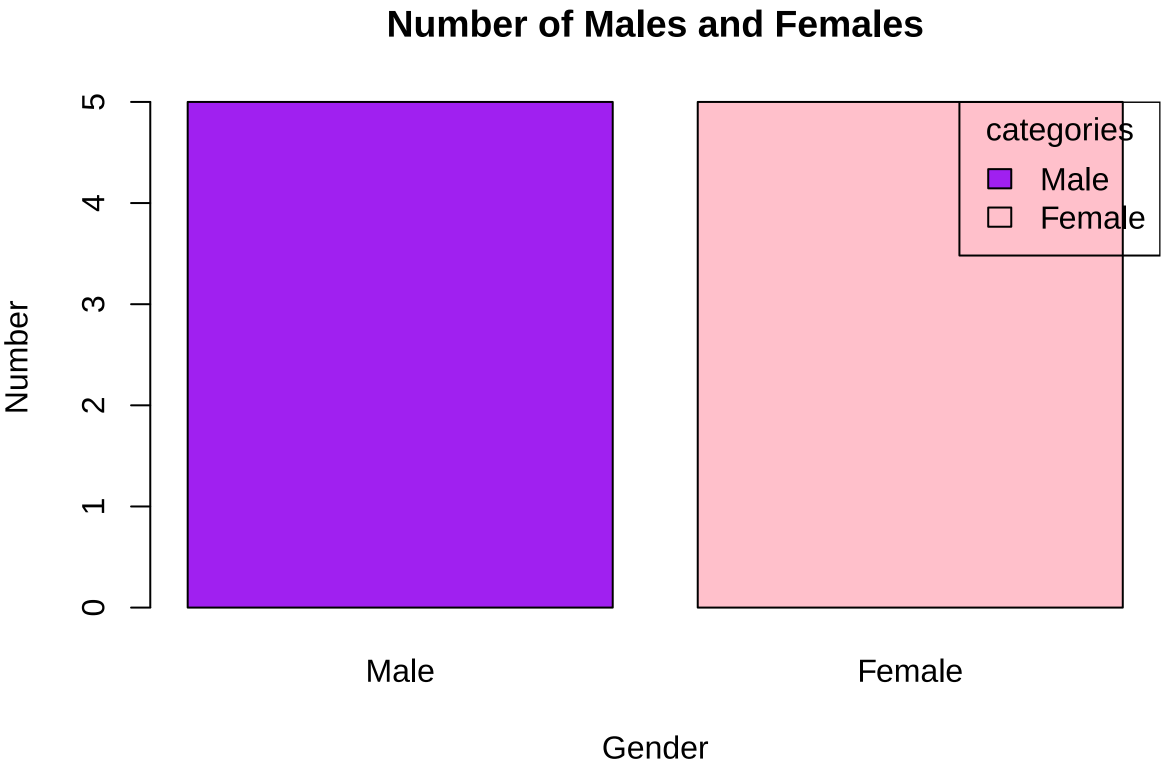
```
## # A tibble: 14 x 3
##    `Shoe Size` Height Gender
##          <dbl>  <dbl> <chr>
##  1         6.5   66   F
##  2         9     68   F
##  3         8.5   64.5 F
##  4         8.5   65   F
##  5         7     64   F
##  6         9.5   70   F
##  7         9     71   F
##  8         7.5   64   F
##  9         8.5   67   F
## 10         8.5   59   F
## 11         5     62   F
## 12         6.5   66   F
## 13         7.5   64   F
## 14         8.5   69   F
```

##c. Create a graph for the number of males and females for Household Data. Use plot(), chart type = barplot. Make sure to place title, legends, and colors. Write the R scripts and its result.

```
library(ggplot2)

HouseholdData <- read_csv("HouseholdData.csv")
```

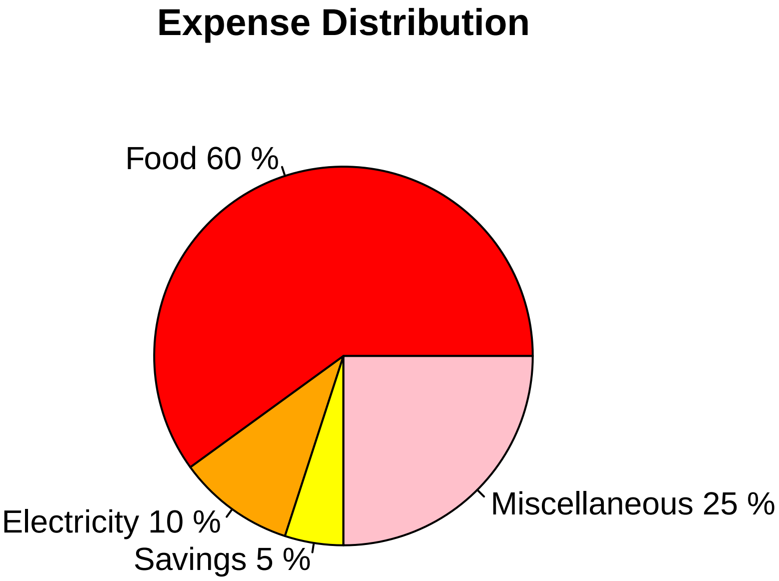
```
## Rows: 10 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Sex, Types of Houses
## dbl (4): Respondents, Fathers Occupation, Persons at Home, Siblings at School
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
barm<- subset(HouseholdData, Sex == "Male")
barf<- subset(HouseholdData, Sex == "Female")
colors<- c("purple","pink")

barplot(
  c(nrow(barm), nrow(barf)),
  names.arg = c("Male", "Female"),
  col = colors,
  main ="Number of Males and Females",
  xlab = "Gender",
  ylab = "Number"
)

legend("topright", legend = c("Male", "Female"), fill =colors, title = "categories")
```



##5. The monthly income of Dela Cruz family was spent. ##a. Create a piechart that will include labels in percentage.Add some colors and title of the chart. Write the R scripts and show its output.

```
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")
values <- c(60, 10, 5, 25)
colors<- c("red","orange","yellow","pink")

pie(values, labels = paste(categories, round(values/sum(values) * 100, 1), "%"), col = colors)

title("Expense Distribution")
```



##6. Use the iris dataset.

```r
data(iris)
```

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

```r
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 ...
```

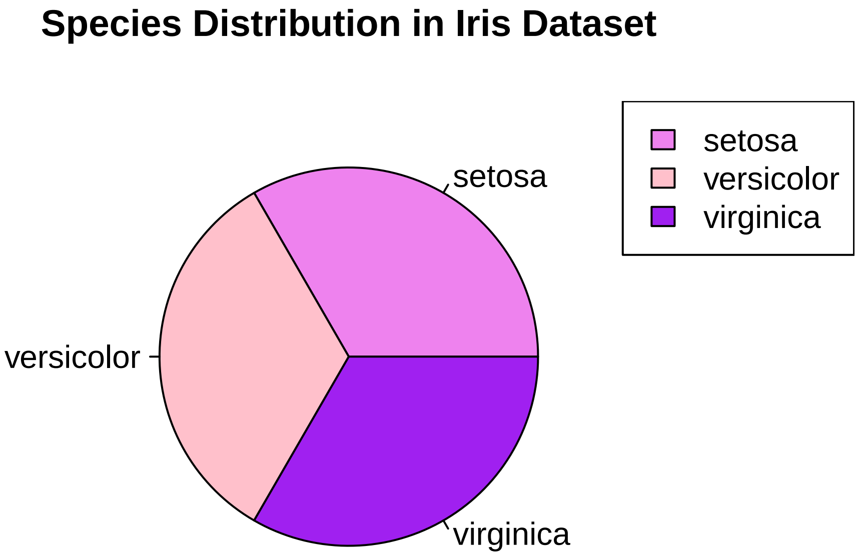
`Str() function provides information about data types and the first few entries in each variable` ##b. Create an R object that will contain the mean of the sepal.length,sepal.width,petal.length,and petal.width. What is the R script and its result?

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

mean_values
```

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

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

```r
species_counts <- table(iris$Species)
colors<- c("violet","pink","purple")

pie(species_counts, labels = levels(iris$Species), col = colors)

title("Species Distribution in Iris Dataset")
legend("topright", legend = levels(iris$Species), fill = colors)
```



##d. Subset the species into setosa, versicolor, and virginica. Write the R script and show the last six (6) rows of each species.

```
setosa_subset <- subset(iris, Species == "setosa")
versicolor_subset <- subset(iris, Species == "versicolor")
virginica_subset <- subset(iris, Species == "virginica")

tail(iris, n=6)
```

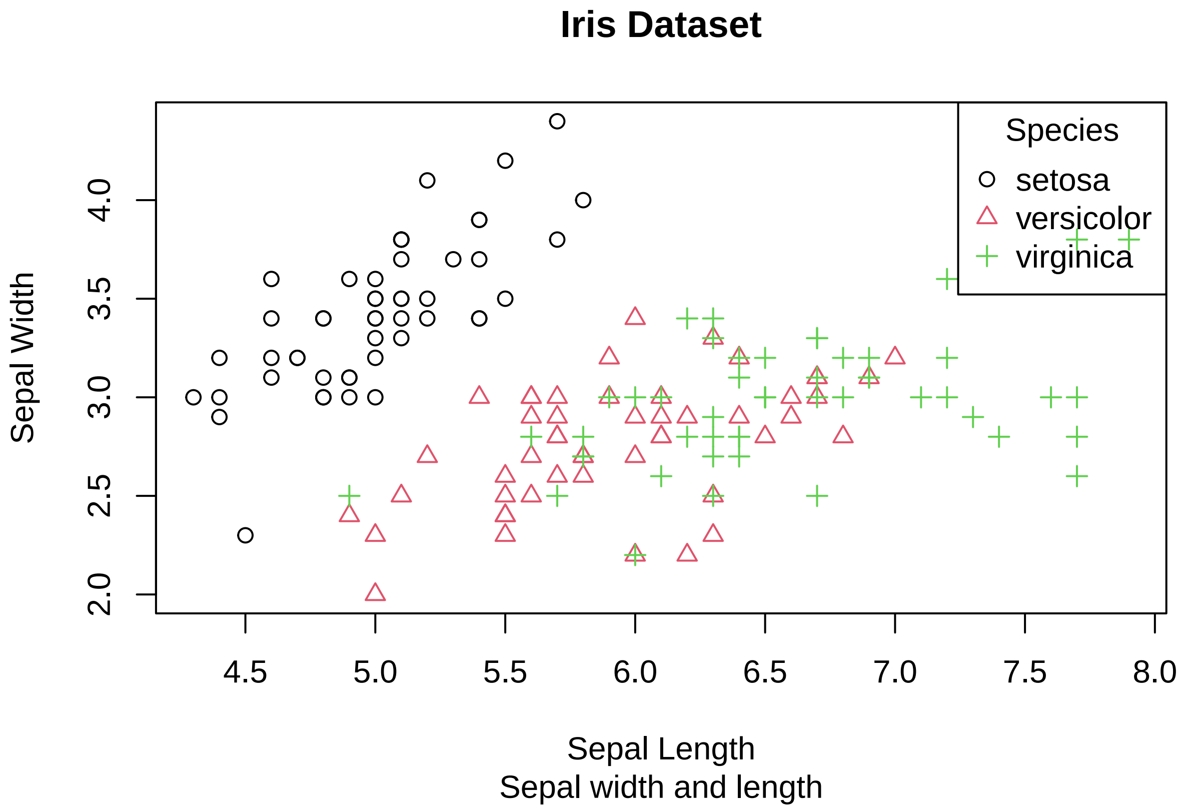
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          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
```

```
setosa_subset
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## 11          5.4         3.7          1.5         0.2  setosa
## 12          4.8         3.4          1.6         0.2  setosa
## 13          4.8         3.0          1.4         0.1  setosa
## 14          4.3         3.0          1.1         0.1  setosa
## 15          5.8         4.0          1.2         0.2  setosa
## 16          5.7         4.4          1.5         0.4  setosa
## 17          5.4         3.9          1.3         0.4  setosa
## 18          5.1         3.5          1.4         0.3  setosa
## 19          5.7         3.8          1.7         0.3  setosa
## 20          5.1         3.8          1.5         0.3  setosa
## 21          5.4         3.4          1.7         0.2  setosa
## 22          5.1         3.7          1.5         0.4  setosa
## 23          4.6         3.6          1.0         0.2  setosa
## 24          5.1         3.3          1.7         0.5  setosa
## 25          4.8         3.4          1.9         0.2  setosa
## 26          5.0         3.0          1.6         0.2  setosa
## 27          5.0         3.4          1.6         0.4  setosa
## 28          5.2         3.5          1.5         0.2  setosa
## 29          5.2         3.4          1.4         0.2  setosa
## 30          4.7         3.2          1.6         0.2  setosa
## 31          4.8         3.1          1.6         0.2  setosa
## 32          5.4         3.4          1.5         0.4  setosa
## 33          5.2         4.1          1.5         0.1  setosa
## 34          5.5         4.2          1.4         0.2  setosa
## 35          4.9         3.1          1.5         0.2  setosa
## 36          5.0         3.2          1.2         0.2  setosa
## 37          5.5         3.5          1.3         0.2  setosa
## 38          4.9         3.6          1.4         0.1  setosa
## 39          4.4         3.0          1.3         0.2  setosa
## 40          5.1         3.4          1.5         0.2  setosa
## 41          5.0         3.5          1.3         0.3  setosa
## 42          4.5         2.3          1.3         0.3  setosa
## 43          4.4         3.2          1.3         0.2  setosa
## 44          5.0         3.5          1.6         0.6  setosa
## 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
```

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.

```r
plot(iris$Sepal.Length, iris$Sepal.Width,
     main = "Iris Dataset",
     sub = "Sepal width and length",
     xlab = "Sepal Length", ylab = "Sepal Width",
     col = as.numeric(iris$Species),
     pch = as.numeric(iris$Species))

legend("topright", legend = levels(iris$Species),
       col = unique(as.numeric(iris$Species)), pch = unique(as.numeric(iris$Species)),
       title = "Species")
```



##f.Interpret the result. Setosa flowers generally have shorter sepal lengths and wider sepal widths, forming a distinct cluster in the lower left part of the plot. However Versicolor flowers are scattered across the middle part of the plot, showing moderate values for both sepal length and width.Lastly, Virginica flowers tend to have longer sepal lengths and varying sepal widths, forming a cluster in the upper part of the plot.