

RWorksheet_Quebral#4b

Myles Andrei Quebral

2024-10-28

#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. Hint Use abs() function to get the absolute value

```
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(i-j)
  }
}

print(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

```
n_rows <- 6

for (i in 1:n_rows) {

  row_pattern <- paste(rep("*", i), collapse = " ")
  print(row_pattern)
}
```

```
## [1] "*"
## [1] "* *"
## [1] "* * *"
## [1] "* * * *"
## [1] "* * * * *"
## [1] "* * * * * *"
```

#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.

```
start_term <- readline(prompt = "Enter the starting term of the Fibonacci sequence: ")

## Enter the starting term of the Fibonacci sequence:
```

```

a <- 0
b <- 1
pos <- 1

repeat {

  fib <- a + b

  a <- b
  b <- fib

  pos <- pos + 1

  if (pos >= start_term) {

    if (fib > 500) {
      break
    }
    print(fib)
  }
}

```

```

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

```

#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

```

setwd("/cloud/project/Rworksheet#4b")
excel <- read.csv("Excel.csv")
excel

```

```

##      Shoe_size Height Gender Shoe_size.1 Height.1 Gender.1
## 1         6.70   66.0     F         13.0       77         M
## 2         9.00   68.0     F         11.5       72         M
## 3         8.50   64.5     F          8.5       59         F
## 4         8.50   65.0     F          5.0       62         F
## 5        10.50   70.0     M         10.0       72         M
## 6         7.00   64.0     F          6.5       66         F
## 7         9.50   70.0     F          7.5       64         M
## 8         9.00   71.0     F          8.5       67         F
## 9        13.00   70.0     M         10.5       73         M
## 10        7.50   64.0     F          8.5       69         F
## 11       10.50   74.5     M         10.5       72         M

```

```
## 12      8.50  67.0    F      11.0    69    F
## 13     12.00  71.0    M       9.0    69    M
## 14     10.18  71.0    M      13.0    70    M
```

```
excel[1:6,]
```

```
##   Shoe_size Height Gender Shoe_size.1 Height.1 Gender.1
## 1      6.7   66.0     F      13.0      77         M
## 2      9.0   68.0     F      11.5      72         M
## 3      8.5   64.5     F       8.5      59         F
## 4      8.5   65.0     F       5.0      62         F
## 5     10.5   70.0     M      10.0      72         M
## 6      7.0   64.0     F       6.5      66         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.

```
male_sub <- subset(excel,Gender=="M")
male_sub
```

```
##   Shoe_size Height Gender Shoe_size.1 Height.1 Gender.1
## 5     10.50   70.0     M      10.0      72         M
## 9     13.00   70.0     M      10.5      73         M
## 11    10.50   74.5     M      10.5      72         M
## 13    12.00   71.0     M       9.0      69         M
## 14    10.18   71.0     M      13.0      70         M
```

```
fem_sub <- subset(excel ,Gender == "F")
fem_sub
```

```
##   Shoe_size Height Gender Shoe_size.1 Height.1 Gender.1
## 1      6.7   66.0     F      13.0      77         M
## 2      9.0   68.0     F      11.5      72         M
## 3      8.5   64.5     F       8.5      59         F
## 4      8.5   65.0     F       5.0      62         F
## 6      7.0   64.0     F       6.5      66         F
## 7      9.5   70.0     F       7.5      64         M
## 8      9.0   71.0     F       8.5      67         F
## 10     7.5   64.0     F       8.5      69         F
## 12     8.5   67.0     F      11.0      69         F
```

```
num_male <- nrow(male_sub)
num_female <- nrow(fem_sub)
num_male
```

```
## [1] 5
```

```
num_female
```

```
## [1] 9
```

#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.

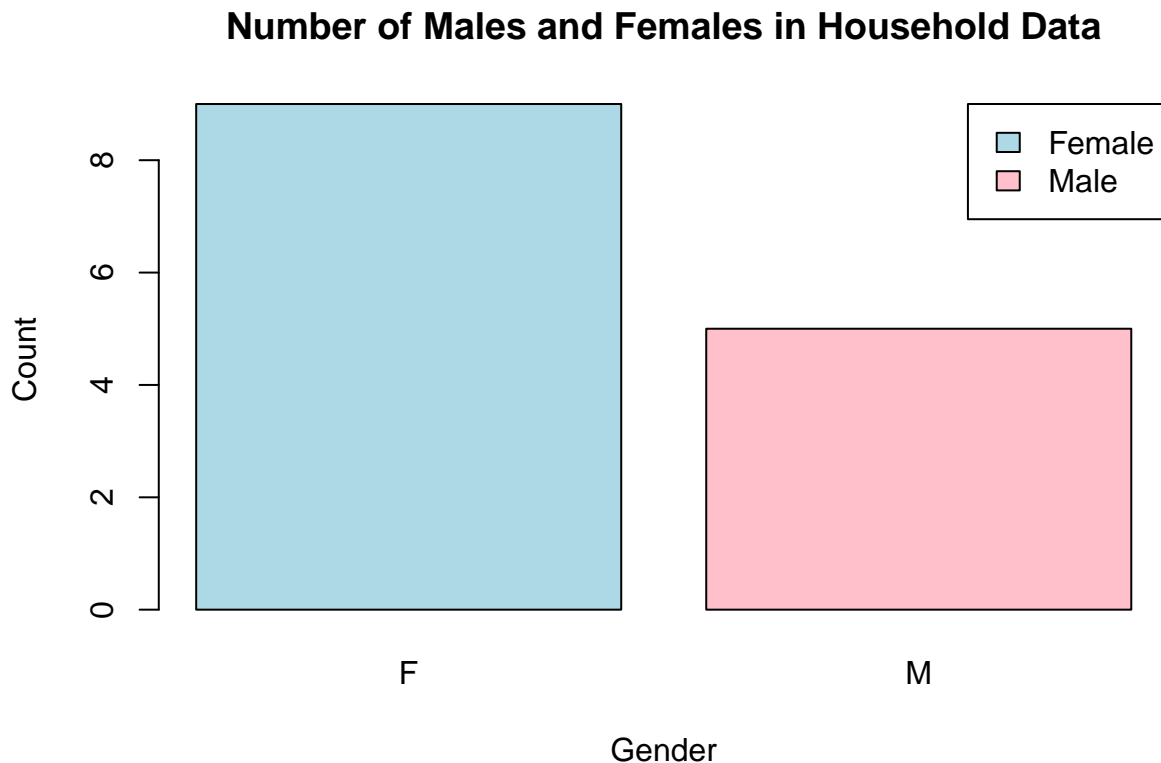
```
gender_counts <- table(excel$Gender)
```

```
barplot(
  gender_counts,
```

```

main = "Number of Males and Females in Household Data",
xlab = "Gender",
ylab = "Count",
col = c("lightblue", "pink"),
legend.text = c("Female", "Male"),
args.legend = list(x = "topright")
)

```



#5.
The monthly income of Dela Cruz family was spent on the following: Food Electricity Savings Miscellaneous
60 10 5 25 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.

```

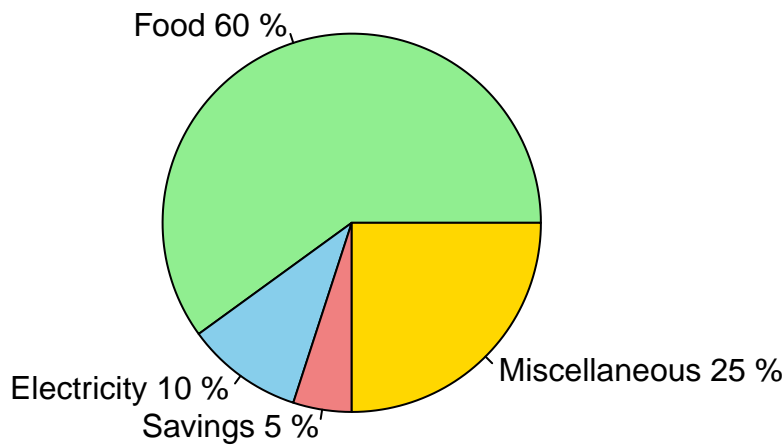
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)

percent_labels <- paste(names(expenses), round(expenses / sum(expenses) * 100), "%")

pie(
  expenses,
  labels = percent_labels,
  col = c("lightgreen", "skyblue", "lightcoral", "gold"),
  main = "Monthly Expenses of Dela Cruz Family"
)

```

Monthly Expenses of Dela Cruz Family



#6. Use the iris dataset.

```
data(iris)

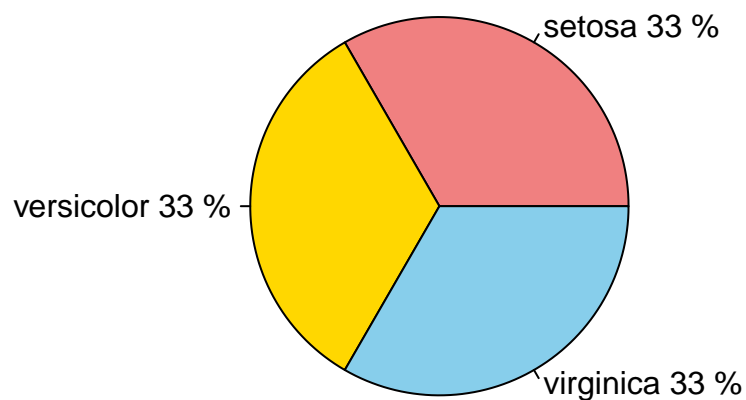
species_counts <- table(iris$Species)

species_percent <- round(species_counts / sum(species_counts) * 100)

colors <- c("lightcoral", "gold", "skyblue")

pie(species_counts,
     labels = paste(names(species_counts), species_percent, "%"),
     col = colors,
     main = "Distribution of Iris Species (Spending Categories)")
```

Distribution of Iris Species (Spending Categories)



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

```
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. 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?

```
mean <- colMeans(iris[, 1:4])
mean
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
## 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.

```
data(iris)

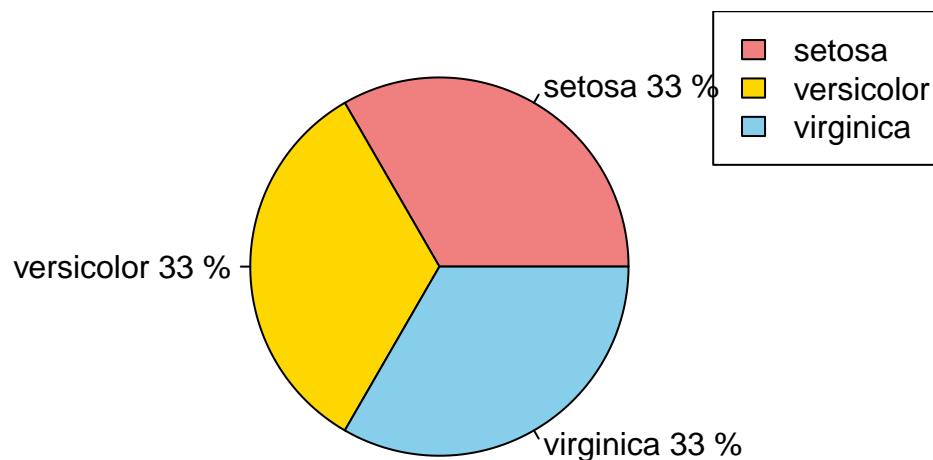
species_counts <- table(iris$Species)

species_percent <- round(species_counts / sum(species_counts) * 100)

colors <- c("lightcoral", "gold", "skyblue")

pie(species_counts,
    labels = paste(names(species_counts), species_percent, "%"),
    col = colors,
    main = "Distribution of Iris Species (Spending Categories)",
    legend("topright", legend = names(species_counts), fill = colors))
```

Distribution of Iris Species (Spending Categories)



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

```

setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(iris, Species == "virginica")

# Display the last six rows of each species
tail(setosa)

##      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

tail(versicolor)

##      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

tail(virginica)

##      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

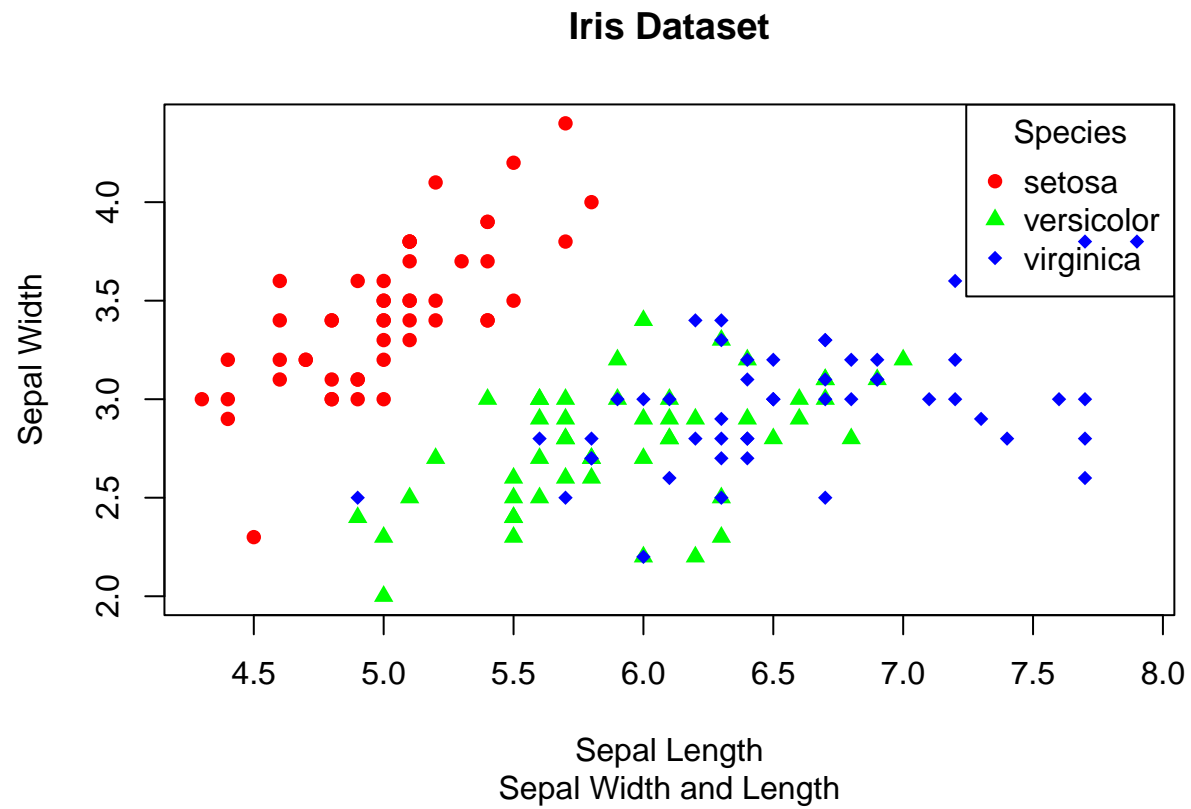
iris$Species <- as.factor(iris$Species)

colors <- c("red", "green", "blue")
symbols <- c(16, 17, 18)

plot(iris$Sepal.Length, iris$Sepal.Width,
     col = colors[iris$Species],          # Colors based on species
     pch = symbols[iris$Species],         # Symbols based on species
     xlab = "Sepal Length",               # Label for x-axis
     ylab = "Sepal Width",               # Label for y-axis
     main = "Iris Dataset",              # Title
     sub = "Sepal Width and Length")     # Subtitle

legend("topright", legend = levels(iris$Species),
     col = colors, pch = symbols,
     title = "Species")

```



#f. Interpret the result # This scatterplot will show the Sepal.Length on the x-axis and Sepal.Width on the y-axis, with points colored and symbolized according to species