RWorksheet_Jacildo#4b

Czharina Mae Jacildo

2024-10-30

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

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

```
##      [,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

```
rows <- 5

for (i in 1:rows) {
  cat(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.

```
user <- (readline(prompt = "Enter a number: "))
```

```
## Enter a number:
```

```
a <- 0
b <- 1

repeat {
  c <- a + b
```

```
  if (c > 500) {
    break
  }

  if (c >= user) {
    print(c)
  }

  a <- b
  b <- c
}
```

```
## [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
```

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(readxl)
shoe_data <- read.table("/cloud/project/worksheet#4/shoe_data.csv", header = TRUE, sep = ",")
```

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.

```
female <- subset(shoe_data, Gender == "F")
male <- subset(shoe_data, Gender == "M")

numF <- nrow(female)
numM <- nrow(male)

print(numF)
```

```
## [1] 14
```

```
print(numM)
```

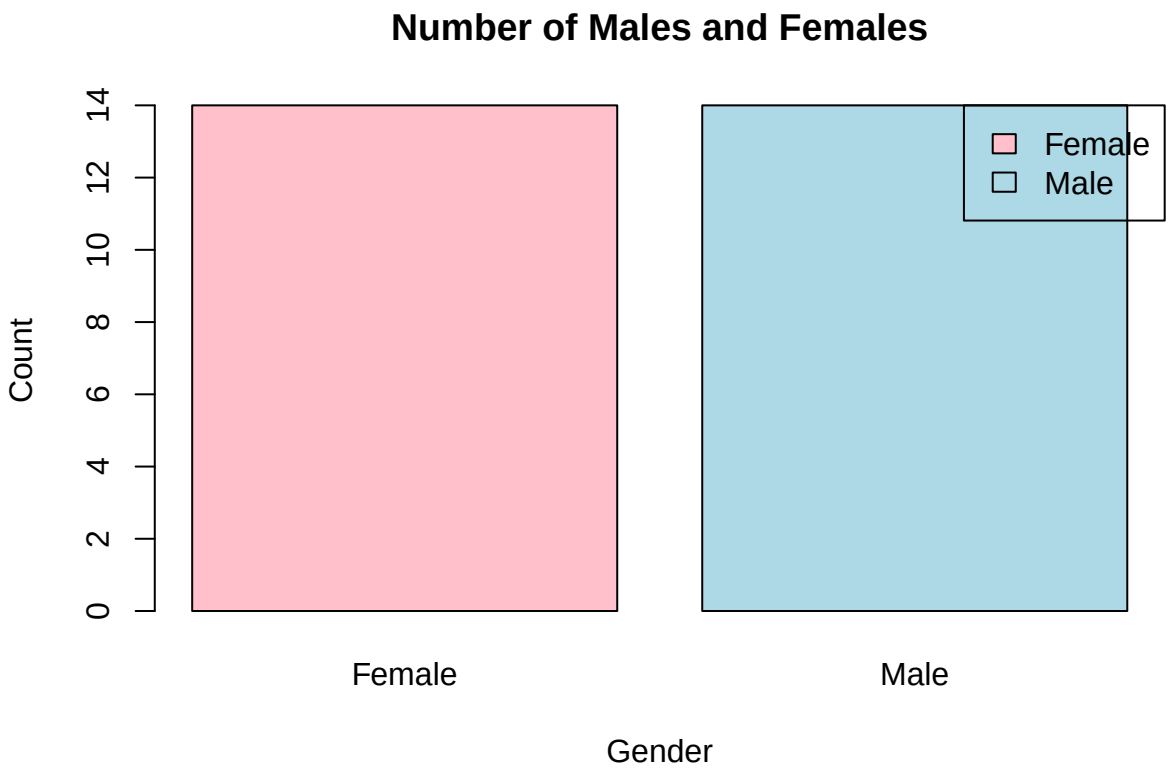
```
## [1] 14
```

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_num <- c(numF, numM)
gender <- c("Female", "Male")

barplot(gender_num, names.arg = gender,
```

```
        col = c("pink", "lightblue"),
        main = "Number of Males and Females",
        xlab = "Gender",
        ylab = "Count",
        legend.text = gender,
        args.legend = list(x = "topright"))
```



5. The monthly income of Dela Cruz family was spent on the following:

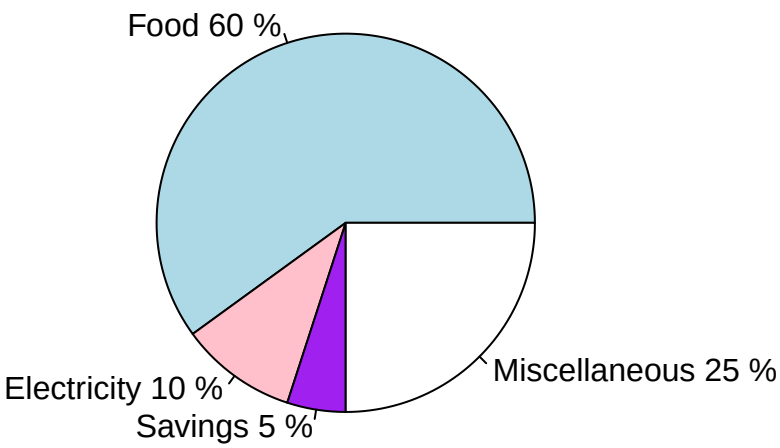
a. Create a piechart that will include labels in percentage.Add some colors and title of the chart. Write the R scripts and show its output.

```
library(ggplot2)

bills <- c(60, 10, 5, 25)
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")

percentage <- round(bills / sum(bills) * 100, 1)
labels <- paste(categories, percentage, "%")

pie(bills, labels = labels,
col = c("lightblue", "pink", "purple", "white"),
main = "Dela Cruz Family Monthly Income")
```



6. Use the iris dataset.

a. Check for the structure of the dataset 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 ...
```

- The output shows the observations and variables of the iris dataframe.

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?

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

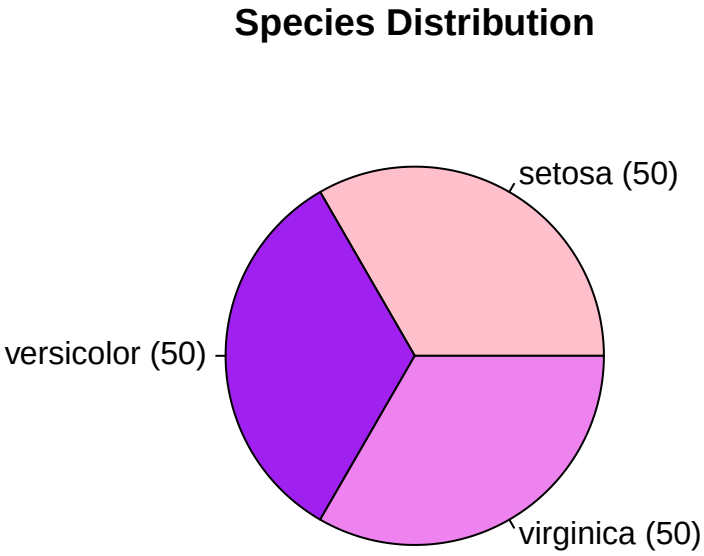
```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

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

```
species_data <- table(iris$Species)

labels <- paste(names(species_data), species_data, sep = " (")
labels <- paste(labels, ")", sep = "")

pie(species_data, labels = labels,
col = c("pink", "purple", "violet"),
main = "Species Distribution"
)
```



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")

tail(setosa, 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
```

```
tail(versicolor, 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
```

```
tail(virginica, 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

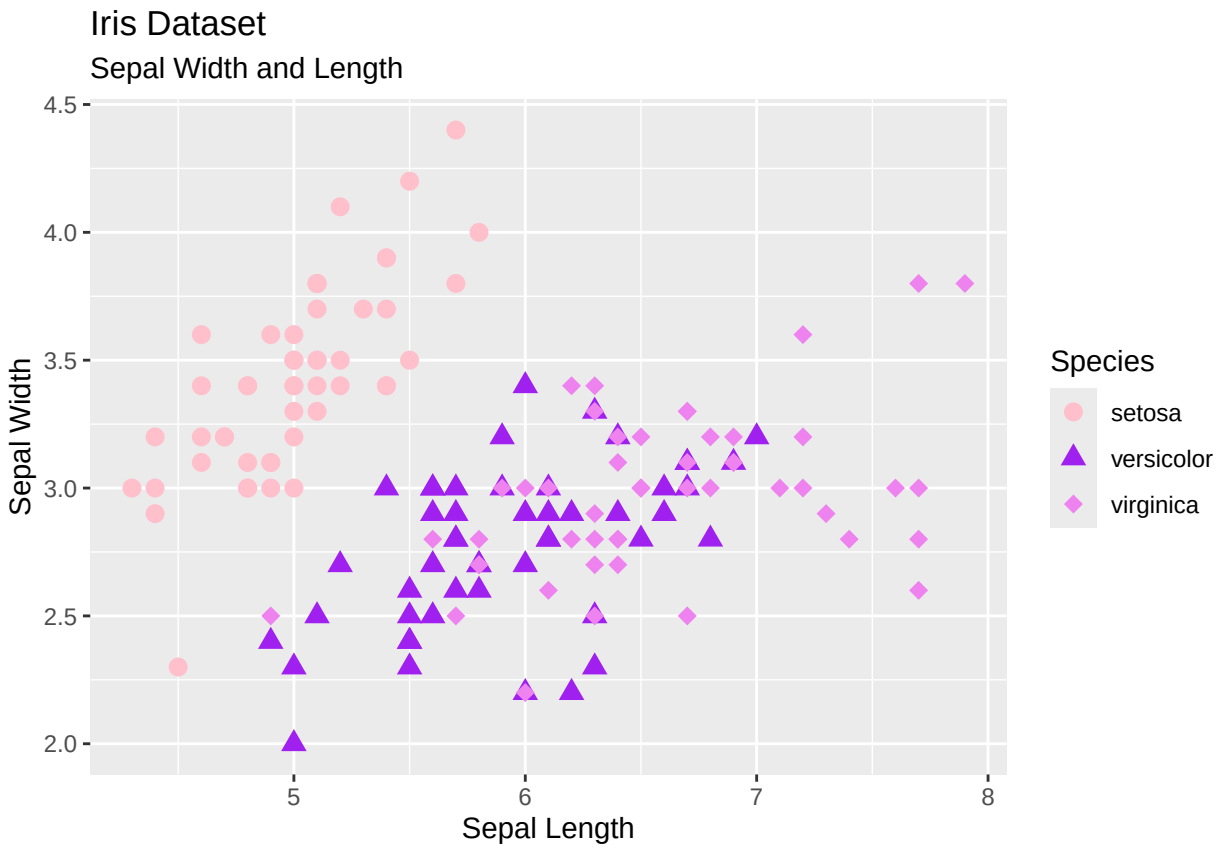
symbol and colors should be based on the species.

```
library(ggplot2)
data(iris)

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

scatter_plot <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species)) +
ggtitle("Iris Dataset") +
labs(subtitle = "Sepal Width and Length", x = "Sepal Length", y = "Sepal Width") +
geom_point(size = 3) +
scale_color_manual(values = c("setosa" = "pink", "versicolor" = "purple", "virginica" = "violet")) +
scale_shape_manual(values = c(16, 17, 18))


print(scatter_plot)
```



f. Interpret the result.

- It shows the Sepal Width and Sepal Length of each species, setosa has most width than length, while versicolor has more length than width, and virginica has most length than width.

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

```
library(readxl)
alexa <- read_excel("/cloud/project/worksheet#4/alexa_file.xlsx")
print(alexa)
```

```
## # A tibble: 3,150 x 5
##    rating date                variation           verified_reviews      feedback
##     <dbl> <dttm>              <chr>               <chr>                    <dbl>
##  1      5 2018-07-31 00:00:00 Charcoal Fabric     Love my Echo!                1
##  2      5 2018-07-31 00:00:00 Charcoal Fabric     Loved it!                    1
##  3      4 2018-07-31 00:00:00 Walnut Finish       Sometimes while play~        1
##  4      5 2018-07-31 00:00:00 Charcoal Fabric     I have had a lot of ~        1
##  5      5 2018-07-31 00:00:00 Charcoal Fabric     Music                        1
##  6      5 2018-07-31 00:00:00 Heather Gray Fabric I received the echo ~        1
##  7      3 2018-07-31 00:00:00 Sandstone Fabric    Without having a cel~        1
##  8      5 2018-07-31 00:00:00 Charcoal Fabric     I think this is the ~        1
##  9      5 2018-07-30 00:00:00 Heather Gray Fabric looks great                  1
## 10      5 2018-07-30 00:00:00 Heather Gray Fabric Love it! I've listen~        1
## # i 3,140 more rows
```

b. Get the total number of each variations and save it into another object. Save the object as variations.RData. Write the R scripts. What is its result?

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
variations <- alexa %>%
count(alexa$variation)

save(variations, file = "/cloud/project/worksheet#4/variations.RData")
load("variations.RData")

print(variations)
```

```
## # A tibble: 16 x 2
##    `alexa$variation`               n
##    <chr>                       <int>
##  1 Black                         261
##  2 Black  Dot                   516
##  3 Black  Plus                  270
##  4 Black  Show                  265
##  5 Black  Spot                  241
##  6 Charcoal Fabric              430
##  7 Configuration: Fire TV Stick 350
##  8 Heather Gray Fabric          157
##  9 Oak Finish                    14
## 10 Sandstone Fabric              90
## 11 Walnut Finish                  9
## 12 White                         91
## 13 White  Dot                   184
## 14 White  Plus                   78
## 15 White  Show                   85
```

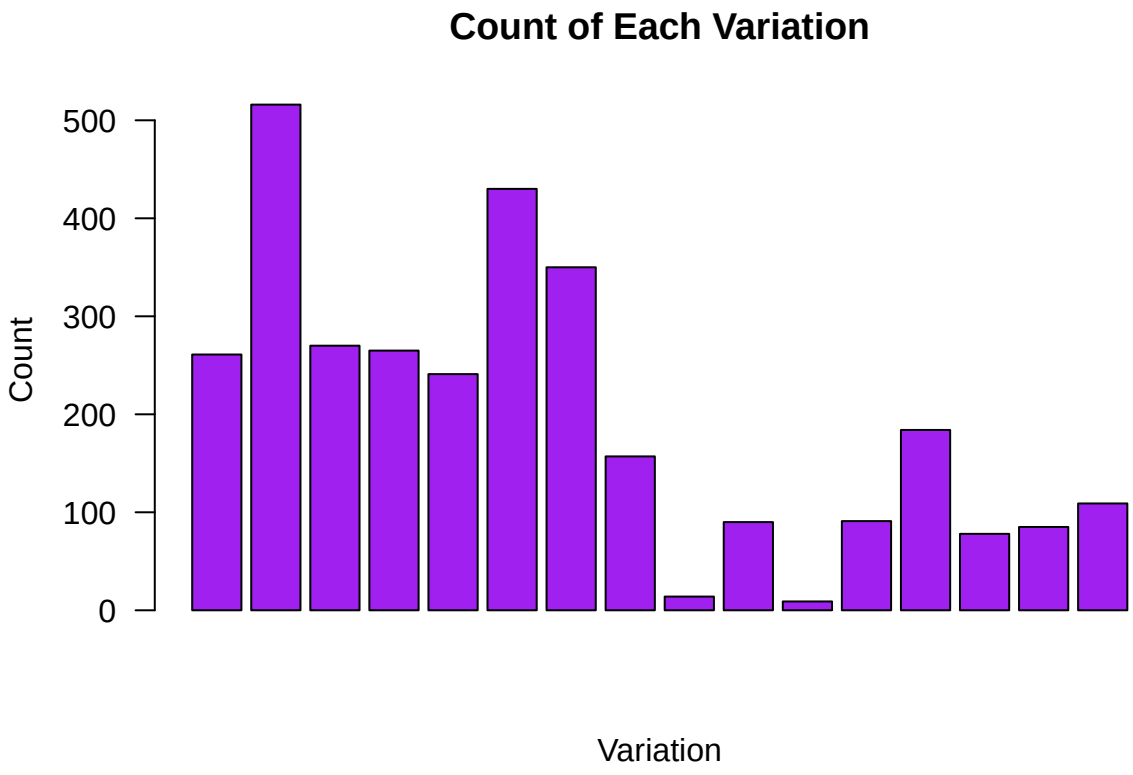
```
## 16 White  Spot                    109
```

c. From the variations.RData, create a barplot(). Complete the details of the chart which include the title, color, labels of each bar.

```
load("variations.RData")

barplot(variations$n,
        names.arg = variations$variation,
        col = "purple",
        main = "Count of Each Variation",
        xlab = "Variation",
        ylab = "Count",
        las = 2)
```

```
## Warning: Unknown or uninitialised column: `variation`.
```



d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the details of the chart.

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
black_var <- data.frame(
  variation = c("Black", "Black Plus", "Black Show", "Black Spot", "Black Dot"),
  Count = c(250, 300, 200, 100, 500)
```

```
)

white_var <- data.frame(
  variation = c("White", "White Dot", "White Plus", "White Show", "White Spot"),
  Count = c(100, 150, 80, 90, 120)
)

plot_black <- ggplot(black_var, aes(x = variation, y = Count, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Black Variants", y = "Variants", x = "Total Numbers") +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.y = element_text(size = 8)
  )

plot_white <- ggplot(white_var, aes(x = variation, y = Count, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "White Variants", y = "Variants", x = "Total Numbers") +
  theme_minimal() +
  theme(
    legend.position = "none",
    axis.text.y = element_text(size = 8)
  )

grid.arrange(plot_black, plot_white, ncol = 2)
```