

# RWorksheet\_Malayas#4b.Rmd

Andrew Miguel M. Malayas BSIT2A

2024-10-30

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

```
vectorA <- c(1, 2, 3, 4, 5)
matrixT <- matrix(0, nrow = 5, ncol = 5)
for (i in 1:5) {
  matrixT[i, i] <- vectorA[i]
}
matrixT
```

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

2. Print the string "\*" using for() function.

```
n <- 5
for (i in 1:n) {
  for (j in 1:i) {
    cat(" * ")
  }
  cat("\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.

```
start <- as.integer(readline(prompt = "Enter the starting number: "))
```

```
## Enter the starting number:
```

```
start <- 1
a <- start
b <- 1
cat(a, b, sep = " ")
```

```
## 1 1
```

```
repeat {
  nxt <- a + b
  if (!is.na(nxt) && nxt > 500) {
    break
  }
  cat(nxt, " ")
  a <- b
  b <- nxt
}
```

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

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

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

```
shoes <- read.csv("shoes.csv")
```

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

```
head(shoes)
```

```
##   Shoesize Height Gender
## 1      6.5   66.0      F
## 2      9.0   68.0      F
## 3      8.5   64.5      F
## 4      8.5   65.0      F
## 5     10.5   70.0      M
## 6      7.0   64.0      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.

```
female_subset <- subset(shoes, Gender == "F")
male_subset <- subset(shoes, Gender == "M")

female_num <- nrow(female_subset)
male_num <- nrow(male_subset)

cat("Female:", female_num, "\n")
```

```
## Female: 14
```

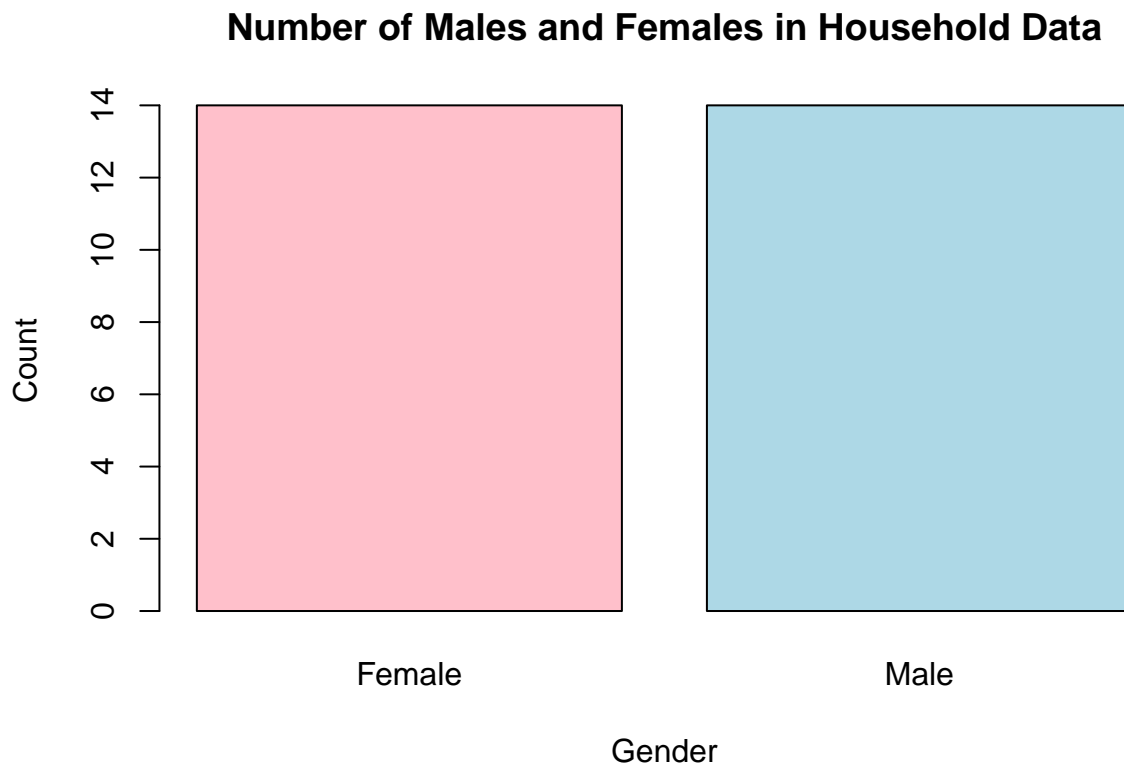
```
cat("Male:", male_num, "\n")
```

```
## Male: 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_counts <- table(shoes$Gender)
barplot(gender_counts,
  main = "Number of Males and Females in Household Data",
  xlab = "Gender",
  ylab = "Count",
  col = c("pink", "lightblue"),

  names.arg = c("Female", "Male"))
```

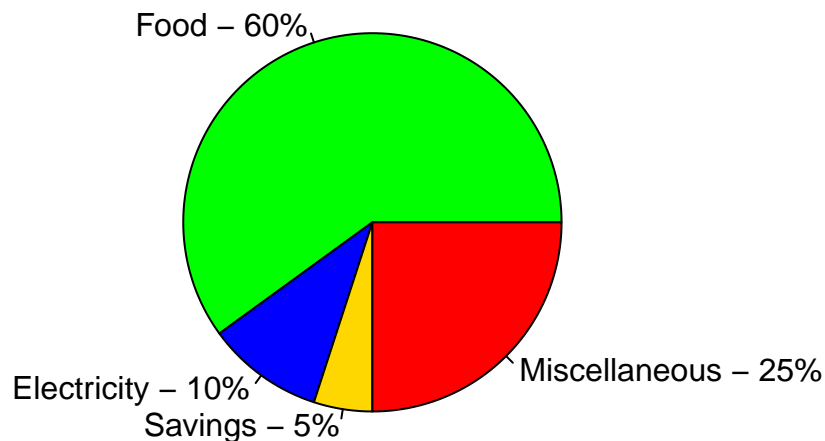


5. The monthly income of Dela Cruz family was spent on the following:

- 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.

```
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
percent_labels <- paste0(names(expenses), " - ", round(expenses / sum(expenses) * 100), "%")
pie(expenses,
    labels = percent_labels,
    col = c("green", "blue", "gold", "red"),
    main = "Dela Cruz Family Monthly Expenses")
```

## Dela Cruz Family Monthly Expenses



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 ...
```

- 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?

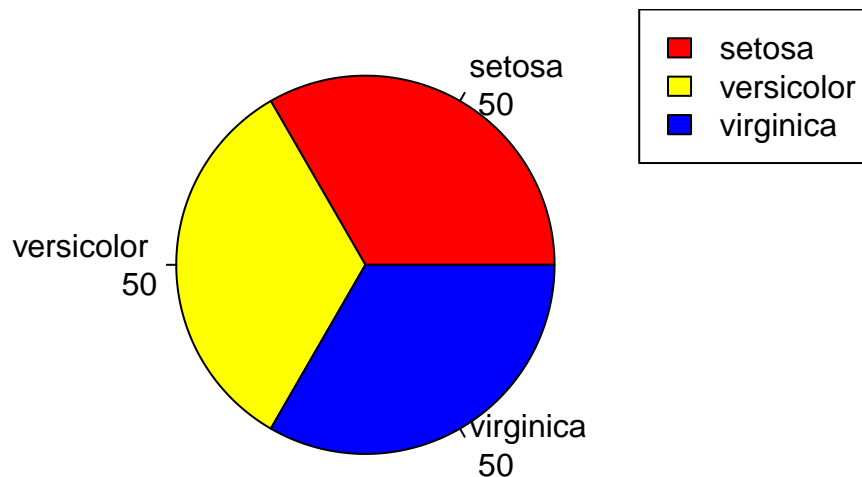
```
colMeans(iris[, 1:4])
```

```
## 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_counts <- table(iris$Species)  
pie(species_counts,  
    main = "Species Distribution in Iris Dataset",  
    col = c("red", "yellow", "blue"),  
    labels = paste(names(species_counts), "\n", species_counts))  
legend("topright",  
    legend = names(species_counts),  
    fill = c("red", "yellow", "blue"))
```

### Species Distribution in Iris Dataset



- 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)
```

```
##      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)
```

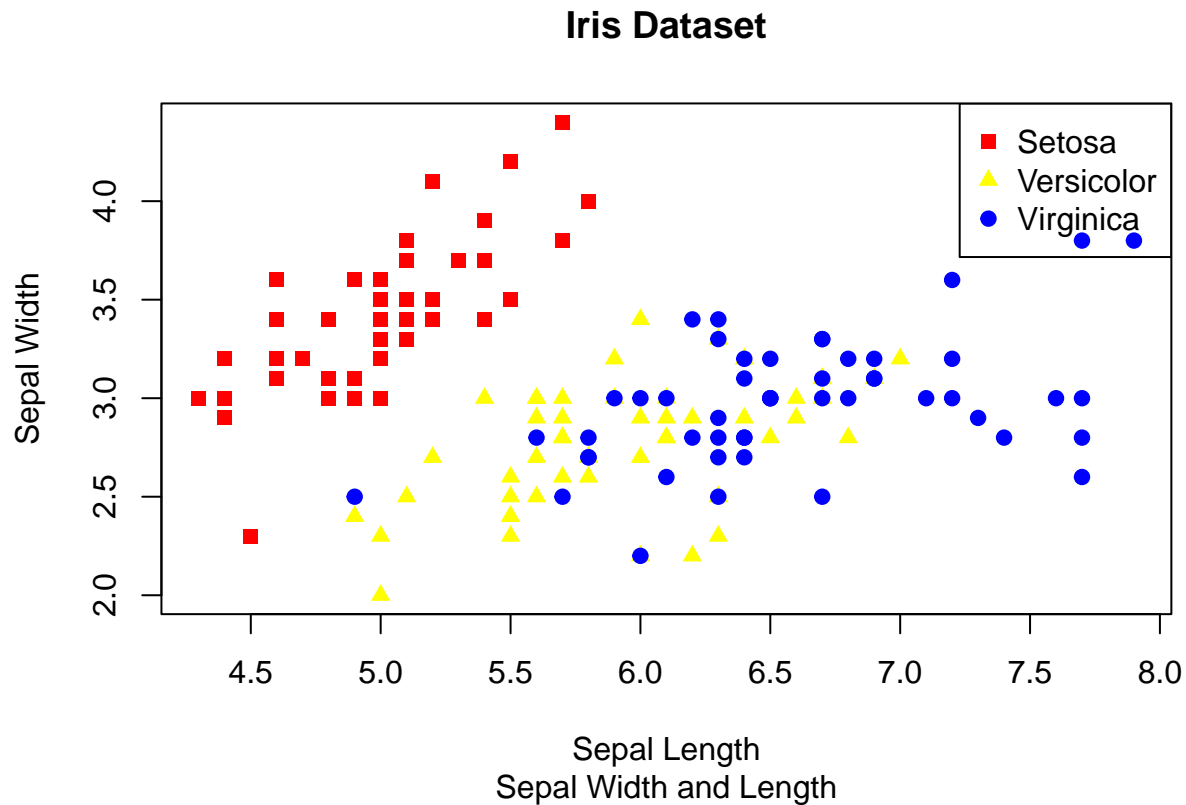
```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 95          5.6         2.7         4.2         1.3 versicolor
## 96          5.7         3.0         4.2         1.2 versicolor
## 97          5.7         2.9         4.2         1.3 versicolor
## 98          6.2         2.9         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
```

- 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.

```
colors <- c("setosa" = "red", "versicolor" = "yellow", "virginica" = "blue")
symbols <- c("setosa" = 15, "versicolor" = 17, "virginica" = 19)
plot(iris$Sepal.Length, iris$Sepal.Width,
     col = colors[iris$Species],
     pch = symbols[iris$Species],
     main = "Iris Dataset",
     sub = "Sepal Width and Length",
     xlab = "Sepal Length",
     ylab = "Sepal Width")
legend("topright", legend = c("Setosa", "Versicolor", "Virginica"),
     col = c("red", "yellow", "blue"),
     pch = c(15, 17, 19))
```



#### f. Overall Interpretation

The scatterplot displays clear groupings for each Iris species based on sepal dimensions:

- Setosa (red squares) features shorter, broader sepals, setting it apart from other species.
- Versicolor (yellow triangles) occupies a middle range with moderate sepal length and width.
- Virginica (blue circles) generally has longer, narrower sepals, with some overlap with versicolor.

This visualization highlights species distinctions, with setosa standing out the most and a slight inverse trend between sepal length and width, particularly in virginica.

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_data <- read_excel("alexa_file.xlsx")
```

```
str(alexa_data)
```

```
## tibble [3,150 x 5] (S3: tbl_df/tbl/data.frame)
##  $ rating      : num [1:3150] 5 5 4 5 5 5 3 5 5 5 ...
##  $ date        : POSIXct[1:3150], format: "2018-07-31" "2018-07-31" ...
```

```
## $ variation      : chr [1:3150] "Charcoal Fabric" "Charcoal Fabric" "Walnut Finish" "Charcoal Fabr
## $ verified_reviews: chr [1:3150] "Love my Echo!" "Loved it!" "Sometimes while playing a game, you c
## $ feedback       : num [1:3150] 1 1 1 1 1 1 1 1 1 1 ...
```