

RWorksheet_Pineda#4b

Epiphany Louise O. Pineda

2025-11-06

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

```
##r

vectorA <- c(1, 2, 3, 4, 5)
matrixResult <- matrix(0, nrow = 5, ncol = 5)

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

print(matrixResult)
```

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

```
for (i in 1:5) {
  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.

```
first <- as.numeric(readline(prompt = "Enter first number: "))
second <- as.numeric(readline(prompt = "Enter second number: "))

cat(first, second, " ")

repeat {
  nextNum <- first + second
  if (nextNum > 500) {
    break
  }
  cat(nextNum, " ")
}
```

```

first <- second
second <- nextNum
}

```

Using Basic Graphics (plot(),barplot(),pie(),hist())

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

```

shoeSize <- read.csv("RWorksheets#4 - #4c/RWorksheet#4b/shoe_size.csv")

head(shoeSize)

```

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(shoeSize, Gender == "F")
male <- subset(shoeSize, Gender == "M")

cat("Number of Female observations:", nrow(female), "\n")
cat("Number of Male observations:", nrow(male), "\n")

```

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.

```

genderCount <- table(shoeSize$Gender)

barplot(genderCount,
  main = "Number of Males and Females in Shoe Size Data",
  xlab = "Gender",
  ylab = "Count",
  col = c("pink", "lightblue"),
  legend.text = c("Female", "Male"))

```

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.

Data

```

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

percent <- round(expenses / sum(expenses) * 100)

labels <- paste(categories, percent, "%")

pie(expenses,
  labels = labels,
  col = c("gold", "skyblue", "lightgreen", "tomato"),
  main = "Monthly Income Distribution of Dela Cruz Family")

```

6.) Use the iris dataset.

```
#data(iris)
```

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

```
data(iris)

str(iris)
```

- 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?

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

irisMeans
```

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

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

pie(speciesCount,
    labels = names(speciesCount),
    col = c("lightgreen", "skyblue", "pink"),
    main = "Distribution of Iris Species")

legend("topright",
    legend = names(speciesCount),
    fill = c("lightgreen", "skyblue", "pink"))
```

- 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)
tail(versicolor)
tail(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. Hint: Need to convert to factors the species to store categorical variables.

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

plot(iris$Sepal.Length, iris$Sepal.Width,
    col = c("red", "green", "blue")[iris$Species],
    pch = 19,
    main = "Iris Dataset",
    sub = "Sepal Width and Length",
    xlab = "Sepal Length (cm)",
    ylab = "Sepal Width (cm)")
```

```
legend("topright",
      legend = levels(iris$Species),
      col = c("red", "green", "blue"),
      pch = 19)
```

f. Interpret the result.

Setosa flowers generally have shorter sepal lengths but wider sepals. Versicolor species lie in the middle range for both measurements. Virginica species have longer sepals and narrower widths. This shows how sepal dimensions can be used to distinguish between iris species.

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, BlackSpot). Also on the white variants (White Dot, White Plus, White Show, White Spot).

```
alexa <- read_excel("RWorksheets#4 - #4c/RWorksheet#4b/alexa_file.xlsx")
```

a. Rename the white and black variants by using gsub() function.

```
alexa$variation <- gsub("Black Dot", "Black Dot", alexa$variation)
alexa$variation <- gsub("Black Plus", "Black Plus", alexa$variation)
alexa$variation <- gsub("Black Show", "Black Show", alexa$variation)
alexa$variation <- gsub("Black Spot", "Black Spot", alexa$variation)

alexa$variation <- gsub("White Dot", "White Dot", alexa$variation)
alexa$variation <- gsub("White Plus", "White Plus", alexa$variation)
alexa$variation <- gsub("White Show", "White Show", alexa$variation)
alexa$variation <- gsub("White Spot", "White Spot", alexa$variation)
```

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? Hint: Use the dplyr package. Make sure to install it before loading the package.

```
library(dplyr)

variations <- alexa %>%
  count(variation)

variations

save(variations, file = "variations.RData")
```

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

```
variationCounts <- variations$n
names(variationCounts) <- variations$variation

barplot(
  variationCounts,
  main = "Total Number of Each Alexa Variation",
  xlab = "Variations",
  ylab = "Total Count",
  col = rainbow(length(variationCounts)),
  las = 2
)
```

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

```
blackVariants <- variations %>%
  filter(grepl("Black", variation))

whiteVariants <- variations %>%
  filter(grepl("White", variation))

blackCounts <- blackVariants$n
names(blackCounts) <- blackVariants$variation

whiteCounts <- whiteVariants$n
names(whiteCounts) <- whiteVariants$variation

par(mfrow = c(1, 2))

barplot(
  blackCounts,
  main = "Black Variants",
  xlab = "Total Numbers",
  ylab = "Values",
  col = c("black", "darkred", "darkgreen", "darkblue", "deepskyblue"),
  las = 2
)

barplot(
  whiteCounts,
  main = "White Variants",
  xlab = "Total Numbers",
  ylab = "Values",
  col = c("gray70", "pink", "lightblue", "turquoise"),
  las = 2
)
```