RWorksheet-3_Amuan#3A

2023-10-13

#USING VECTORS

#1. There is a built-in vector LETTERS contains the uppercase letters of the alphabet and letters which contains the lowercase letters of the alphabet.

#LETTERS

```
print(LETTERS)
print(letters)
```

#1.A. You need to produce a vector that contains the first 11 letters.

```
aVector <- LETTERS [1:11]
aVector
```

#OUTPUT = [1] “A” “B” “C” “D” “E” “F” “G” “H” “I” “J” “K”

#1.B. Produce a vector that contains the odd numbered letters.

```
odd_numbers <- c(1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25)

aVector <- LETTERS [odd_numbers]
aVector
```

#OUTPUT = [1] “A” “C” “E” “G” “I” “K” “M” “O” “Q” “S” “U” “W” “Y”

#1.C. Produce a vector that contains the vowels

```
vowels <- c(1,5,9,15)

aVector <- LETTERS [vowels]
aVector
```

#OUTPUT = [1] “A” “E” “I” “O”

#1.D. Produce a vector that contains the last 5 lowercase letters.

```
aVector <- letters[21:26]
aVector
```

#OUTPUT = [1] “u” “v” “w” “x” “y” “z”

#1.E. Produce a vector that contains letters between 15 to 24 letters in lowercase.

```
aVector <- letters[15:24]
aVector
```

#OUTPUT = [1] “o” “p” “q” “r” “s” “t” “u” “v” “w” “x”

#Create a vector(not a dataframe) with the average temperatures in April for Tugue-garao City, Manila, Iloilo City, Tacloban, Samal Island, and Davao City. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees.

#2.A What is the R code and its result for creating a character vector for the city/town?

```
Cities <- c("Tuguegarao City", "Manila", "Iloilo City",                "Tacloban", "Samal Island", "Dava
Cities
```

#OUTPUT: [1] “Tuguegarao City” “Manila” “Iloilo City” “Tacloban”
#[5] “Samal Island” “Davao City”

#2.B The average temperatures.

```
Temp <- c(42,39,34,34,30,27)
Temp
```

#OUTPUT: [1] 42 39 34 34 30 27

#3.C Create a dataframe to combine the city and the temp.

```
cityTemp <- data.frame(Cities, Temp)
cityTemp
```

#OUTPUT: Cities Temp #1 Tuguegarao City 42 #2 Manila 39 #3 Iloilo City 34 #4 Tacloban 34 #5 Samal Island 30 #6 Davao City 27

#3.D Associate the dataframe you have created in 2.

```
  names(Cities) <- c("Cities", "Temperature")
  cityTemp
```

#OUTPUT: Cities Temp #1 Tuguegarao City 42 #2 Manila 39 #3 Iloilo City 34 #4 Tacloban 34 #5 Samal Island 30 #6 Davao City 27

#3.E Print the structure by using str() function. Describe the output.

```
  str(cityTemp)
```

#DESCRIPTION & OUTPUT: #‘data.frame’: 6 obs. of 2 variables:, This shows that the data frame have 6 observation or known as row and 2 columnns. # $ Cities: chr “Tuguegarao City” “Manila” “Iloilo City” “Tacloban” ... #$ Temp : num 42 39 34 34 30 27 #This shows the Temperature column and indicate that it is a numeric variable.

#3.F From the answer in d, what is the content of row 3 and row 4 What is its R code and its output?

```
  content <- cityTemp[3:4,]
  content
```

#OUTPUT: Cities Temp #3 Iloilo City 34 #4 Tacloban 34

#3.G From the answer in d, display the city with highest temperature and the city with the lowest temperature. What is its R code and its output?

#lowesTemperature

```
    lowestTemp <- cityTemp[which.max(cityTemp$Temp),]
    lowestTemp
```

#OUTPUT: Cities Temp #1 Tuguegarao City 42

#highestTemperature

```
    highestTemp <- cityTemp[which.max(cityTemp$Temp),]
    highestTemp
```

#OUTPUT: Cities Temp #1 Tuguegarao City 42

#USING MATRICES #2. Create a matrix of one to eight and eleven to fourteen with four columns and three rows. #2.A What will be the R code for the #2 question and its result?

```
  matrix <- matrix (c(1:8, 11:14), nrow = 3, ncol = 4)
  matrix
```

#OUTPUT: [,1] [,2] [,3] [,4] #[1,] 1 4 7 12 #[2,] 2 5 8 13 #[3,] 3 6 11 14

#2.B Multiply the matrix by two. What is its R code and its result?

```
  matrixTwo <- matrix * 2
  matrixTwo
```

#OUTPUT: [,1] [,2] [,3] [,4] #[1,] 2 8 14 24 #[2,] 4 10 16 26 #[3,] 6 12 22 28

#2. c. What is the content of row 2? What is its R code?

```
  matrixTwo[2]
```

#OUTPUT: [1] 4

#2. D. What will be the R code if you want to display the column 3 and column 4 in row 1 and row 2? What is its output?

```
  matrixTwo [1:2, 3:4]
```

#OUTPUT:[,1] [,2] #[1,] 14 24 #[2,] 16 26

#2.E What is the R code is you want to display only the columns in 2 and 3, row 3? What is its output?

```
matrixTwo[3, 2:3]
```

#OUTPUT: [1] 12 22

#2.F What is the R code is you want to display only the columns 4? What is its output?

```
matrixTwo[,4]
```

#OUTPUT: [1] 24 26 28

#2.G Name the rows as isa, dalawa, tatlo and columns as uno, dos, tres, quatro for the matrix that was created in b.∴ #What is its R code and corresponding output?

```
dimnames(matrixTwo) <- list(c("isa", "dalawa", "tatlo"), c("uno",         "dos", "tres", "quatro"))
matrixTwo
```

#OUTPUT: uno dos tres quatro #isa 2 8 14 24 #dalawa 4 10 16 26 #tatlo 6 12 22 28

#2.H From the original matrix you have created in a, reshape the matrix by assigning a new dimension with dim(). New dimensions should have 2 columns and 6 rows. #What will be the R code and its output?

```
dim(matrix) <- c(6,2)
matrix
```

#OUTPUT: [,1] [,2] #[1,] 1 7 #[2,] 2 8 #[3,] 3 11 #[4,] 4 12 #[5,] 5 13 #[6,] 6 14

#USING ARRAYS

#3. An array contains 1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1 #3. A Create an array for the above numeric values. Each values will be repeated twice #What will be the R code if you are to create a three-dimensional array with 4 columns and 2 rows. What will be its output?

```
numVal <- c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1)
```

#TWICE

```
  repTwice <- rep(numVal, each = 2)
  repTwice

  array <- array(repTwice, dim = c(2,4,3))
```

```
array
```

#OUTPUT: , , 1

#[,1] [,2] [,3] [,4] #[1,] 1 2 3 6 #[2,] 1 2 3 6

#, , 2

#[,1] [,2] [,3] [,4] #[1,] 7 8 9 0 #[2,] 7 8 9 0

#, , 3

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

#2.B How many dimensions do your array have? # IT HAS THREE (3) DIMENSIONS.

#Name the rows as lowercase letters and columns as uppercase letters starting from the A. The array names should be "1st-Dimensional Array", "2nd-Dimensional Array", and "3rd-Dimensional Array". #What will be the R codes and its output?

```
dimnames(array) <- list(
 row_names <- letters[1:2],
 col_names <- LETTERS[1:4], c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional Arra
array
```

#OUPUT: #, , 1st-Dimensional Array

#A B C D #a 1 2 3 6 #b 1 2 3 6

#, , 2nd-Dimensional Array

#A B C D #a 7 8 9 0 #b 7 8 9 0

#, , 3rd-Dimensional Array

#A B C D #a 3 4 5 1 #b 3 4 5 1

"'