

RWorksheet_Camarista#3a

John Lyxton Camarista

2024-10-02

Using Vectors

```
LETTERS
```

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.

```
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"  
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
letters
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"  
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
#a. You need to produce a vector that contains the first 11 letters  
LETTERS[1:11]
```

Based on the above vector **LETTERS**:

```
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K"
```

```
#b. Produce a vector that contains the odd numbered letters.  
odd_LETTERS <- LETTERS[seq(1, 26, by = 2)]  
odd_LETTERS
```

```
## [1] "A" "C" "E" "G" "I" "K" "M" "O" "Q" "S" "U" "W" "Y"
```

```
#c. Produce a vector that contains the vowels  
vowels <- c("A", "E", "I", "O", "U" )
```

```
#d. Produce a vector that contains the last 5 lowercase letters.
letters[22:26]
```

Based on the above vector letters:

```
## [1] "v" "w" "x" "y" "z"
```

```
#e. Produce a vector that contains letters between 15 to 24 letters in lowercase.
letters[15:24]
```

```
## [1] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
```

2. Create a vector(not a dataframe) with the average temperatures in April for Tuguegarao City, Manila, Iloilo City, Tacloban, Samal Island, and Davao City.

```
#a. What is the R code and its result for creating a character vector for the city/town of Tuguegarao C
city <- c("Tuguegarao City", "Manila", "Iloilo City", "Tacloban City", "Samal Island", "Davao City")
city
```

The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees

```
## [1] "Tuguegarao City" "Manila"           "Iloilo City"      "Tacloban City"
## [5] "Samal Island"    "Davao City"
```

```
#b. The average temperatures in Celsius are 42, 39, 34, 34, 30, and 27 degrees. Name the object as temp
temp <- c(42, 39, 34, 34, 30, 27)
temp
```

```
## [1] 42 39 34 34 30 27
```

```
#c. Create a dataframe to combine the city and the temp by using 'data.frame()'. What the R code and its
CityTemp <- data.frame(City = city, Temp = temp)
CityTemp
```

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

```
#d. Associate the dataframe you have created in 2.(c) by naming the columns using the names() function.
names(CityTemp) <- c("City", "Temperature")
CityTemp
```

```
##           City Temperature
## 1 Tuguegarao City         42
## 2           Manila         39
## 3      Iloilo City         34
## 4  Tacloban City          34
## 5      Samal Island        30
## 6      Davao City          27
```

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

```
str(CityTemp)
```

```
## 'data.frame':   6 obs. of  2 variables:
## $ City      : chr  "Tuguegarao City" "Manila" "Iloilo City" "Tacloban City" ...
## $ Temperature: num  42 39 34 34 30 27
```

- The output displays the number of objects and variables. It also displays the data type of each vectors, the column names and their values.

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

```
CityTemp[3:4, ]
```

```
##           City Temperature
## 3      Iloilo City         34
## 4  Tacloban City          34
```

#g. From the answer in d, display the city with highest temperature and the city with the lowest temperature

```
CityHighTemp <- CityTemp[which.max(CityTemp$Temperature), "City"]
CityLowTemp <- CityTemp[which.min(CityTemp$Temperature), "City"]
print(paste(CityHighTemp, "has the highest temperature."))
```

```
## [1] "Tuguegarao City has the highest temperature."
```

```
print(paste(CityLowTemp, "has the lowest temperature."))
```

```
## [1] "Davao City has the lowest temperature."
```

Using Matrices

- Matrix can be created by specifying the rows and columns

#- a. What will be the R code for the #2 question and its result?

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

2. Create a matrix of one to eight and eleven to fourteen with four columns and three rows.

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

#- b. Multiply the matrix by two. What is its R code and its result?

```
matrixMultiplied <- mat * 2
matrixMultiplied
```

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

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

```
mat[2, ]
```

```
## [1]  2  5  8 13
```

#- 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

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

```
##      [,1] [,2]
## [1,]    7   12
## [2,]    8   13
```

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

```
mat[3, 2:3]
```

```
## [1]  6 11
```

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

```
mat[, 4]
```

```
## [1] 12 13 14
```

#- g. Name the rows as isa, dalawa, tatlo and columns as uno, dos, tres, quatro for the matrix that was

```
rownames(matrixMultiplied) <- c("isa", "dalawa", "tatlo")
colnames(matrixMultiplied) <- c("isa", "dalawa", "tatlo", "quatro")
matrixMultiplied
```

```
##      isa dalawa tatlo quatro
## isa      2      8     14     24
## dalawa   4     10     16     26
## tatlo    6     12     22     28
```

#- h. From the original matrix you have created in a, reshape the matrix by assigning a new dimension w

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

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

Using Arrays

- Array can have more than two dimensions by using the `array()` function and `dim()` to specify the dimensions

```
#- a. Create an array for the above numeric values. Each values will be repeated twice. What will be the dimensions?
arr <- array(rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), times = 2), dim = c(2, 4, 3))
arr
```

3. An array contains 1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    7    9
## [2,]    2    6    8    0
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]    3    5    1    3
## [2,]    4    1    2    6
##
## , , 3
##
##      [,1] [,2] [,3] [,4]
## [1,]    7    9    3    5
## [2,]    8    0    4    1
```

```
#- b. How many dimensions do your array have?
count <- length(dim(arr))
print(paste("My array has", count, "dimensions"))
```

```
## [1] "My array has 3 dimensions"
```

```
#- c. Name the rows as lowercase letters and columns as uppercase letters starting from the A. The array will be 2x4x3.
dimnames(arr) <- list(
  letters[1:2], LETTERS[1:4], c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional Array")
)
arr
```

```
## , , 1st-Dimensional Array
##
##   A B C D
## a 1 3 7 9
## b 2 6 8 0
##
## , , 2nd-Dimensional Array
##
##   A B C D
## a 3 5 1 3
## b 4 1 2 6
##
## , , 3rd-Dimensional Array
##
##   A B C D
## a 7 9 3 5
## b 8 0 4 1
```