

Worksheet#2

ANGELO A. GARRIDO

2025-10-13

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

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

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

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

#b. Produce a vector that contains the odd-numbered letters

```
odd_num_letters <- LETTERS[seq(1, 26, by = 2)]
odd_num_letters
```

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

#c. Produce a vector that contains the vowel

```
vowels <- LETTERS[c(1, 5, 9, 15, 21)]
vowels
```

```
## [1] "A" "E" "I" "O" "U"
```

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

```
last5 <- letters[22:26]
last5
```

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

#e. Produce a vector that contains letters between 15 to 24 (lowercase).

```
letters_Between15_24 <- letters[15:24]
letters_Between15_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 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

#a. Character vector for the cities

```
city <- c("Tuguegarao City", "Manila", "Iloilo City", "Tacloban", "Samal Island", "Davao City")
city
```

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

```
#b. Numeric vector for the temperature
```

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

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

```
#c. Data Frame City and Temp
```

```
city_temp <- data.frame(city, temp)
city_temp
```

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

```
#d. Rename the columns using names() function
```

```
names(city_temp) <- c("City", "Temperature")
city_temp
```

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

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

```
str(city_temp)
```

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

```
#f. Display the content of row 3 and row 4
```

```
city_temp[3, ]
```

```
##           City Temperature
## 3 Iloilo City         34
```

```
city_temp[4, ]
```

```
##           City Temperature
## 4 Tacloban         34
```

```
#g. Display the city with the highest and lowest temperature
```

```
# Highest temperature
```

```
city_temp[which.max(city_temp$Temperature), ]
```

```
##           City Temperature
## 1 Tuguegarao City         42
```

```
# Lowest temperature
```

```
city_temp[which.min(city_temp$Temperature), ]
```

```
##           City Temperature
## 6 Davao City           27
```

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

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

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

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

#result: matrix has 3 rows and 4 columns. numbers 1 to 8 and 11 to 14 are filled column-wise by default.

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

```
matrix_data * 2
```

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

#result: Each Element in the matrix was multiplied by

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

```
matrix_data[2, ]
```

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

#result: Row 2 contains the values 2, 5, 8, and 13.

#d. Display column 3 and column 4 in row 1 and row 2

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

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

#Output: 7, 12

8, 13

#e. Display only columns 2 and 3 in row 3

```
matrix_data[3, 2:3]
```

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

#Output: 6, 11

```
#f. Display only column 4
matrix_data[, 4]
```

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

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

```
#g. Name the rows and columns
```

```
matrix_2 <- matrix_data * 2
rownames(matrix_2) <- c("isa", "dalawa", "tatlo")
colnames(matrix_2) <- c("uno", "dos", "tres", "quatro")
matrix_2
```

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

```
#Output: uno dos tres quatro
```

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

```
#h. Reshaping the Matrix
```

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

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

```
#Output:
```

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

Using Arrays

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

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

```
#a. Create a three-dimensional array
array_data <- array(rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), 2),
                    dim = c(2, 4, 3))
```

```
array_data
```

```
## , , 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
```

```
#output:
```

```
#      [,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 does the array have?
```

```
length(dim(array_data))
```

```
## [1] 3
```

```
#output: 3
```

```
#c. Name the rows, columns, and array layers
```

```
rownames(array_data) <- c("a", "b")
colnames(array_data) <- c("A", "B", "C", "D")
dimnames(array_data) <- list(c("a", "b"),
                             c("A", "B", "C", "D"),
                             c("1st-Dimensional Array",
                               "2nd-Dimensional Array",
```

```
                                "3rd-Dimensional Array"))  
array_data
```

```
## , , 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
```

```
#output:
```

```
#, , 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
```