

RWorksheet_Perez#3a

2024-09-30

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`

```
## [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"
```

Based on the above vector `LETTERS`:

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

```
eleven <- c(LETTERS[1:11])
eleven
```

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

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

```
odd <- LETTERS[seq(1, length(LETTERS), by = 2)]
odd
```

```
## [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")
vowels
```

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

Based on the above vector `letters`:

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

```
last_five <- tail(letters, 5)
last_five
```

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

- e. Produce a vector that contains letters between 15 to 24 letters in lowercase.

```
fifteen_twentyfour <- c(letters[15:24])
fifteen_twentyfour
```

```
## [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. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees.

```
avg_temp <- c(Tuguegarao = 42, Manila = 39, Iloilo = 34, Tacloban = 34, Samal_Island = 30, Davao = 27)
avg_temp
```

```
##      Tuguegarao      Manila      Iloilo      Tacloban Samal_Island      Davao
##           42           39           34           34           30           27
```

- a. What is the R code and its result for creating a character vector for the city/town of Tuguegarao City, Manila, Iloilo City, Tacloban, Samal Island, and Davao City? Name the object as city. The names should follow the same order as in the instruction.

```
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. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees. Name the object as temp. Write the R code and its output. Numbers should also follow what is in the instruction.

```
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 result?

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

```
##           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. Associate the dataframe you have created in 2.(c) by naming the columns using the names() function. Change the column names by using names() function as City and Temperature. What is the R code and its result?

```
names(city_temp_df)
```

```
## [1] "city" "temp"
```

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

```
##           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_df)
```

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

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

```
city_temp_df[3:4, ]
```

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

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?

```
highest <- city_temp_df[which.max(city_temp_df$Temperature), ]
highest
```

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

```
lowest <- city_temp_df[which.min(city_temp_df$Temperature), ]
lowest
```

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

Using Matrices

- Matrix can be created by specifying the rows and columns.

row = 2

```
matrix(c(5,6,7,4,3,2,1,2,3,7,8,9),nrow = 2)
```

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

row = 3 and column = 2

```
matrix(data = c(3,4,5,6,7,8),3,2)
```

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

creating a diagonal matrix where x value will always be 1

```
diag(1,nrow = 6,ncol = 5)
```

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

```
diag(6)
```

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

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

```
##      [,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?

```
matrix_two <- matrix*2
matrix_two
```

```
##      [,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?

```
matrix[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 is its output?

```
column_row <- matrix[1:2,3:4]
column_row
```

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

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

```
col_row <- matrix[3, 2:3]
col_row
```

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

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

```
column <- matrix[, 4]
column
```

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

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?

```
rownames(matrix) <- c("isa", "dalawa", "tatlo")
colnames(matrix) <- c("uno", "dos", "tres", "quatro")
matrix
```

```
##      uno dos tres quatro
## isa      1  4   7    12
## dalawa   2  5   8    13
## tatlo    3  6  11    14
```

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
```

```
##      [,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.

creates a two-dimensional array containing numbers from 1 to 24 that have 3 rows and 4 columns

```
array_dta <- array(c(1:24), c(3,4,2))
array_dta
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
##
## , , 2
##
##      [,1] [,2] [,3] [,4]
## [1,]   13   16   19   22
## [2,]   14   17   20   23
## [3,]   15   18   21   24
```

checking for the dimensions

row, column, dimension

```
dim(array_dta)
```

```
## [1] 3 4 2
```

```
#checking for the number of elements
```

```
length(array_dta)
```

```
## [1] 24
```

- Another way to create arrays

```
vectorA <- c(1:24)
```

creating an array

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

```
an_Array
```

```
## , , 1
```

```
##
```

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

```
## [1,]    1    4    7   10
```

```
## [2,]    2    5    8   11
```

```
## [3,]    3    6    9   12
```

```
##
```

```
## , , 2
```

```
##
```

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

```
## [1,]   13   16   19   22
```

```
## [2,]   14   17   20   23
```

```
## [3,]   15   18   21   24
```

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

- 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?

```
numbers <- rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), times = 2)
```

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

```
numeric_array
```

```
## , , 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?

```
dim(numeric_array)
```

```
## [1] 2 4 3
```

c. 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(numeric_array) <- list(c("a", "b"), c("A", "B", "C", "D"), c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional Array"))
numeric_array
```

```
## , , 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
```