

# RWorksheet\_Cahutay#3a

Mark Anton Cahutay

2024-10-02

Using Vectors 1.

```
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"
```

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

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

```
## [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, length(LETTERS), 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 <- LETTERS[LETTERS %in% c("A", "E", "I", "O", "U")]  
vowels
```

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

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

```
last_five <- letters[(length(letters) - 4):length(letters)]  
last_five
```

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

```
#E. Produce a vector that contains letters between 15 to 24 letters in lowercase.
my_small_letters <- letters[15:24]
my_small_letters
```

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

2. Create a vector with the average temperatures in April

```
#A. Vector for city names
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. Vector for city's average temperature
temp <- c(42, 39, 34, 34, 30, 27)
temp
```

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

```
#C. Data Frame for the city and average temperature
citytemp_df <- data.frame(city, temp)
citytemp_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. Modify column names of Data Frame
names(citytemp_df) <- c("City", "Temperature")
citytemp_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.
citytemp_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
```

```
str(citytemp_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. The content of row 3 and row 4 from the answer in D.
```

```
my_rows <- citytemp_df[c(3, 4), ]
my_rows
```

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

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

```
highest_temp <- max(citytemp_df$Temperature)
lowest_temp <- min(citytemp_df$Temperature)
city_highest <- citytemp_df[citytemp_df$Temperature == highest_temp, ]
city_highest
```

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

```
city_lowest <- citytemp_df[citytemp_df$Temperature == lowest_temp, ]
city_lowest
```

```
##           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. R code and its result
```

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

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

*#B. R code and result of matrix multiplied by two*

```
matrix2 <- my_matrix * 2  
matrix2
```

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

*#C. R code for accessing the content in row 2*

```
my_matrix[2,]
```

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

*#D. R code to display content of column 3 and 4 in row 1 and 2*

```
my_matrix[(c(1, 2)), (c(3,4))]
```

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

*#E. R code to display only the columns in 2 and 3, row 3.*

```
my_matrix[3, (c(2,3))]
```

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

*#F. R code to display only the columns 4.*

```
my_matrix[, 4]
```

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

*#G. Naming the rows and columns of matrix created from B.*

```
dimnames(matrix2) <- list(c("isa", "dalawa", "tatlo"), c("uno", "dos", "tres", "quatro"))  
matrix2
```

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

*#H. Assigning new dimension of the matrix created from A.*

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

```
##      [,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

```
#A. Create an array for the above values
```

```
my_array <- c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1)
my_3Darray <- array(rep(my_array, each = 2), dim = c(2,4,3))
my_3Darray
```

```
## , , 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
```

```
#B. The number of dimensions my array have
```

```
dim(my_3Darray)
```

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

```
#C.
```

```
dimnames(my_3Darray) <- list(
  c("a", "b"),
  c("A", "B", "c", "D"),
  c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional Array")
)
my_3Darray
```

```
## , , 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
```