RWorksheet_Pineda#3a

Epiphany Louise O. Pineda

2025-10-13

```r
#Using Vectors
#(1.) There is a built-in vector LETTERS contains the uppercase letters of the alphabet and letters whi
#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.
first_11 <- LETTERS[1:11]
first_11
#Output:[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
#Output:[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
#Output:[1] "A" "E" "I" "O" "U"

#Based on the above vector letters:
#d. Produce a vector that contains the last 5 lowercase letters.
last_5 <- letters[22:26]
last_5
#Output:[1] "v" "w" "x" "y" "z"

#e. Produce a vector that contains letters between 15 to 24 letters in lowercase.
range_15_24 <- letters[15:24]
range_15_24
#Output:[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, Man

#a. What is the R code and its result for creating a character vector for the city/town of Tuguegarao C
```

```r
city <- c("Tuguegarao City", "Manila", "Iloilo City", "Tacloban", "Samal Island", "Davao City")
city
#Output:[1] "Tuguegarao City" "Manila" "Iloilo City" "Tacloban" "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 in
temp <- c(42, 39, 34, 34, 30, 27)
temp
#Output:[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
temperature_data <- data.frame(city, temp)
temperature_data
#Output:           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 usingthe names() function.
names(temperature_data) <- c("City", "Temperature")
temperature_data
#Output:
#      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(temperature_data)
#Output: The output shows that temperature_data is a data frame with 6 observations (rows) and 2 variab
#'data.frame':   6 obs. of  2 variables:
#$ City        : chr  "Tuguegarao City" "Manila" "Iloilo City" "Tacloban" "Samal Island" "Davao City"
#$ 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?
temperature_data[3:4, ]
#Output:
#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?
temperature_data[which.max(temperature_data$Temperature), ]
#Output:
#Highest temperature:
#temperature_data[which.max(temperature_data$Temperature), ]
```

```r
#Output:
#  City              Temperature
#1 Tuguegarao City        42

#Lowest temperature:
#temperature_data[which.min(temperature_data$Temperature), ]
#Output:
#  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?
m <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
m
#Output:
#     [,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?
m2 <- m * 2
m2
#Output:
#      [,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?
m[2, ]
#Output:[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 i.
m[1:2, 3:4]
#Output:
#      [,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?
m[3, 2:3]
#Output:[1]   6 11

#f. What is the R code is you want to display only the columns 4? What is its output?
m[, 4]
#Output:[1] 12 13 14

#g. Name the rows as isa, dalawa, tatlo and columns as uno, dos, tres, quatro for the matrix that was c
rownames(m2) <- c("isa", "dalawa", "tatlo")
colnames(m2) <- c("uno", "dos", "tres", "quatro")
m2
#Output:
#        uno dos tres 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 wit.
dim(m) <- c(6, 2)
m
#Output:
#      [,1] [,2]
#[1,]    1    7
#[2,]    2   12
#[3,]    3   13
#[4,]    4   14
#[5,]    5    8
#[6,]    6   11

#Using Arrays
```

```r
#• Array can have more than two dimensions by using the array() function and dim() to specify the dimen

# 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
```

```r
# checking for the dimensions
# row, column, dimension
dim(array_dta)
```

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

```r
#checking for the number of elements
length(array_dta)
```

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

```r
#• 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
```

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

```r
values_rep <- rep(values, 2)

my_array <- array(values_rep, dim = c(2, 4, 3))
my_array
#Output:
#, , 1

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


#, , 2

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


#, , 3

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


#b. How many dimensions do your array have?
dim(my_array)
#Output:[1] 2 4 3

#c. Name the rows as lowercase letters and columns as uppercase letters starting from the A. The array
rownames(my_array) <- c("a", "b")
colnames(my_array) <- c("A", "B", "C", "D")

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

my_array
#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 1
#b  4 7 8 9


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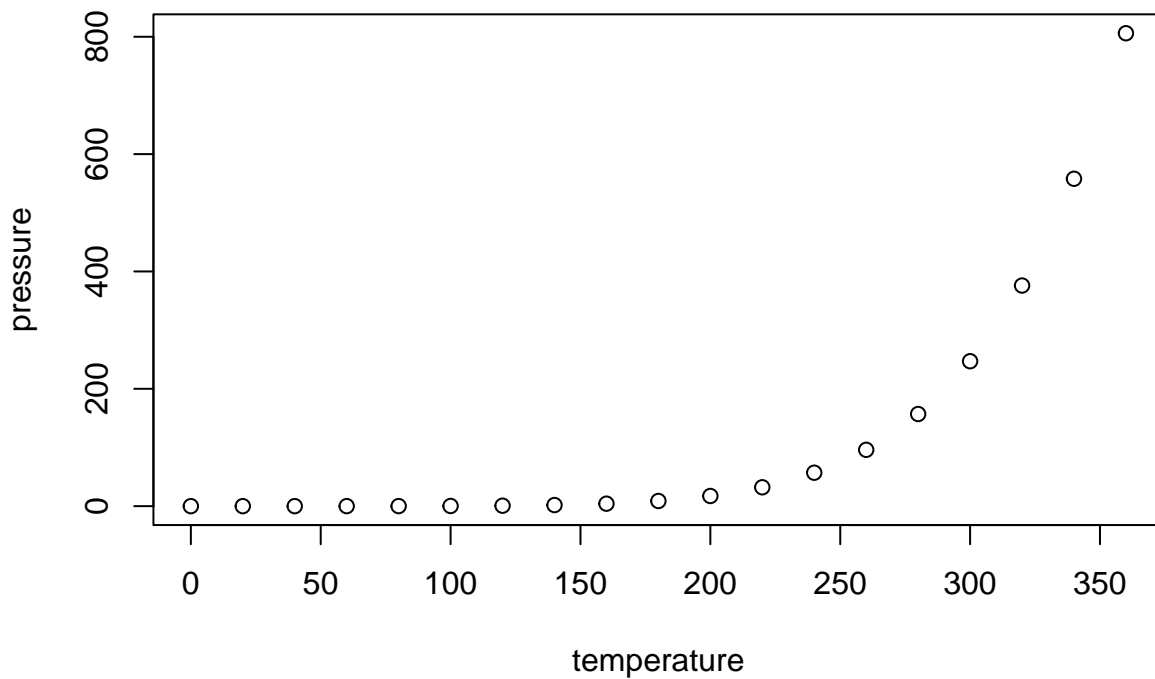
```
#   A B C D
#a  0 3 4 5
#b  1 2 6 7
```

```r
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.