WorkSheet3

Carl Frederick Delicana, BSIT 2-A

10-23-2022

```
#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.
#Based on the above vector LETTERS:
#a. You need to produce a vector that contains the first 11 letters.
eleven_Letters <- LETTERS[1:11]
eleven_Letters
#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[1:26 %% 2 != 0]
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
vowel1 <- LETTERS[c(1,5,9,15,21)]
vowel1
#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_Five <- letters[22:26]
last_Five
#Output:
#[1] "v" "w" "x" "y" "z"


#e. Produce a vector that contains letters between 15 to 24 letters in lowercase.
between15_to24 <- letters[15:24]
between15_to24
#Output:
#[1] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"

#2. Create a vector 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.
#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
#Output:
#[1] "Tuguegarao City" "Manila"         "Iloilo City"
#[4] "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
instruction.
temp <- c( 42, 39, 34, 34, 30, 27)
temp
#Output:
#[1] 42 39 34 34 30 27


#c. Associate the temperature temp with the city by using
names() function. What is the R code and its result?
names(temp) <- city
temp
#Output:
#Tuguegarao City         Manila    Iloilo City        Tacloban
#            42             39             34              34
#  Samal Island     Davao City
#            30             27


#e. From the answer in d, what is the content of index 5 and index 6? What is its R code?
indexFive_Six <- temp[5:6]
indexFive_Six
#Output:
#Samal Island    Davao City
#          30            27



#Using Matrices
#• Matrix can be created by specifying the rows and columns.
#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?
number2 <- matrix(c(1:8,11:14),nrow = 3,ncol = 4)
number2
#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?
byTwo <- number2 * 2
byTwo
#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?
rowTwo <- byTwo[2]
rowTwo
#Output:
#[1] 4

#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?
display1 <- byTwo[1:2,3:4]
display1
#Output:
#      [,1] [,2]
#[1,]   14   24
#[2,]   16   26

#e. What is the R code is you want to display only the columns in 2 and 3, row 3?
What is its output?
display2 <- byTwo[3,2:3]
display2
#Output:
#[1] 12 22

#f. What is the R code is you want to display only the columns 4? What is its output?
display3 <- byTwo[,4]
display3
#Output:
#[1] 24 26 28

#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?
dimnames(byTwo) <- list(c("isa","dalawa","tatlo"),c("uno","dos","tres","quatro"))
byTwo
#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 with dim(). New dimensions should have 2 columns and 6 rows.
What will be the R code and its output?
dim(number2) <- c(6,2)
number2
dim(number2)
#Output:
#      [,1] [,2]
#[1,]    1    7
#[2,]    2    8
#[3,]    3   11
#[4,]    4   12
#[5,]    5   13
#[6,]    6   14
```

```
#> dim(number2)
#[1] 6 2

#Using Arrays
#• Array can have more than two dimensions by using the array() function and dim()
to specify the dimensions
#• Another way to create arrays
#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?
elements1 <- array(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1))
elements1
elements2 <- rep(elements1, 2)
elements2
dim(elements2) <- c(2,4,3)
elements2
#Output:
#[1] 1 2 3 6 7 8 9 0 3 4 5 1
#> elements2 <- rep(elements1, 2)
#> elements2
#[1] 1 2 3 6 7 8 9 0 3 4 5 1 1 2 3 6 7 8 9 0 3 4 5 1
#> dim(elements2) <- c(2,4,3)
#> elements2
#, , 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(elements2)
#Output:
#[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-Dimentional Array", and
"3rd-Dimensional Array". What will be the R codes and its output?
dimnames(elements2) <- list(c("a","b"),c("A","B","C","D"),c("1st-Dimensional Array",
"2nd-Dimentional Array","3rd-Dimensional Array"))
elements2
#Output:
```

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