Rworksheet_Canonicato#3

Dianah Marie Canonicato

2023-10-04

```
#Worksheet-3a in R

#* 1. There is a built-in vector LETTERS contains the uppercase letters of the alphabetand letters whic


uppercase_letters <- LETTERS[1:26]
uppercase_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"
```

```
lowercase_letters <- letters [1:26]
lowercase_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.

first11_Upper <-head(LETTERS, 11)
first11_Upper
```

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

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

odd_numbered_letters <- LETTERS[c(TRUE, FALSE)]
odd_numbered_letters
```

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

```
#c. Produce a vector that contains the vowels

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

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

```
#Based on the above vector letters:
#d. Produce a vector that contains the last 5 lowercase letters.

lowercase_5Let <-tail(letters, 5)
lowercase_5Let
```

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

```
# e. Produce a vector that contains letters between 15 to 24 letters in lowercase.
lowercase_15to24 <-letters [c(15:24)]
lowercase_15to24
```

```
##  [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, Manil

#a.
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"
```

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



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

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

```
#Output: 42 39 34 34 30 27

#c.
Temp_city <- data.frame(
  temp = c(42, 39, 34, 34, 30, 27),
  city = c("Tuguegarao City", "Manila", "Iloilo City", "Tacloban", "Samal Island", "Davao City")
)
Temp_city
```

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

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



#d.

names(Temp_city)[c(1, 2)] <- c("Temperature", "City")
colnames
```

```
## function (x, do.NULL = TRUE, prefix = "col")
## {
##     if (is.data.frame(x) && do.NULL)
##         return(names(x))
##     dn <- dimnames(x)
##     if (!is.null(dn[[2L]]))
##         dn[[2L]]
##     else {
##         nc <- NCOL(x)
##         if (do.NULL)
##             NULL
##         else if (nc > 0L)
##             paste0(prefix, seq_len(nc))
##         else character()
##     }
## }
## <bytecode: 0x55f51b042468>
## <environment: namespace:base>
```

```
#Output:
#function (x, do.NULL = TRUE, prefix = "col")
#{
# if (is.data.frame(x) && do.NULL)
#  return(names(x))
# dn <- dimnames(x)
#  if (!is.null(dn[[2L]]))
#    dn[[2L]]
#  else {
#    nc <- NCOL(x)
#    if (do.NULL)
#      NULL
#    else if (nc > 0L)
#      paste0(prefix, seq_len(nc))
#    else character()
#  }
#}
#<bytecode: 0x0000025f68980a78>
#  <environment: namespace:base>
#  >

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


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

str(Temp_city)
```

```
## 'data.frame':    6 obs. of  2 variables:
```

```
##  $ Temperature: num  42 39 34 34 30 27
##  $ City       : chr  "Tuguegarao City" "Manila" "Iloilo City" "Tacloban" ...
```

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

#Temp_city is a data frame with two columns: "Temperature" and "City." "Temperature" contains numeric v


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

colnames(Temp_city)
```

```
## [1] "Temperature" "City"
```

```
Temp_city[3:4,]
```

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

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

#g. From the answer in d, display the city with highest temperature and the city with the lowest temper
max_temp_city <- Temp_city[which.max(Temp_city$Temperature), "City"]
min_temp_city <- Temp_city[which.min(Temp_city$Temperature), "City"]

max_temp_city
```

```
## [1] "Tuguegarao City"
```

```
min_temp_city
```

```
## [1] "Davao City"
```

```
#Output:
#max_temp_city
#[1] "Tuguegarao City"
# min_temp_city
#[1] "Davao City"



#                      Using Matrices



#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(c(1:8, 11:14), ncol=4, nrow=3, )
```

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

```
#Result:
#       [,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?
2 * matrix(c(1:8, 11:14), ncol=4, nrow=3)
```

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

```
#Result:
#       [,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(c(1:8, 11:14), ncol=4, nrow=3)[2,]
```

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

```
#Output: 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
matrix(c(1:8, 11:14), nrow = 3, ncol = 4)[1:2, 3:4]
```

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

```
#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?
matrix(c(1:8, 11:14), nrow = 3, ncol = 4)[3, 2:3]
```

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

```
#Output: 6 11


#f. What is the R code is you want to display only the columns 4? What is its output?
matrix(c(1:8, 11:14), nrow = 3, ncol = 4)[, 4]
```

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

```
#Output: 12 13 14


#g. Name the rows as isa, dalawa, tatlo and columns as uno, dos, tres, quatro for the matrix that was c
mat <- 2* matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
```

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

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

```
#corresponding 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 wi
new_mat <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
dim(new_mat) <- c(6, 2)
new_mat
```

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

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


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

#a.
data <- c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1)

Array_Val <- array (c (1:3, 6:9, 0, 3:5, 1), c (2,4,3))
Array_Val
```

```
## , , 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

#      [,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.
dim(Array_Val)
```

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

```
#Output: 2 4 3

#c.

data <- c(1:3, 6:9, 0, 3:5, 1)
Array_Val <- array(data, dim = c(2, 4, 3))

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

Array_Val
```

```
## , , 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
```

“‘