Worksheet-3a in R

Necel Kate Noblezada

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.

```
first11 <- LETTERS[1:11]
print(first11)
```

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

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

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

```
##  [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")]
print(vowels)
```

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

Based on the above vector letters:

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

```
last5 <- letters[(length(letters)-4):length(letters)]
print(last5)
```

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

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

```
letters15to24 <- letters[15:24]
print(letters15to24)
```

```
##  [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.

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

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

This code creates the data dataframe containing the cities and their average temperatures.

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(data) <- c("City", "Temperature")
print(data)
```

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

The code changes the column names of the dataframe to “City” and “Temperature.”

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

```
str(data)
```

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

The output shows that the dataframe contains 6 rows and 2 columns, with the `City` column as character data and the Temperature column as numeric data.

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

```
rows_3_and_4 <- data[3:4, ]
print(rows_3_and_4)
```

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

The code displays the content of row 3 and row 4, showing the city names and their corresponding temperatures.

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?

```
highesttemp <- data[data$Temperature == max(data$Temperature), ]
lowesttemp <- data[data$Temperature == min(data$Temperature), ]
print(highesttemp)
```

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

```
print(lowesttemp)
```

```
##         City Temperature
## 6 Davao City          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?

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

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

This code creates a matrix with the specified values and structure.

b. Multiply the matrix by two. What is its R code and its result?

```
matrixmul <- matrixdata * 2
print(matrixmul)
```

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

This code multiplies each element of the matrix by 2.

c. What is the content of row 2? What is its R code?

```
matrixdata <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4, byrow = TRUE)
matrixmul <- matrixdata * 2
row2content <- matrixmul[2, ]
print(row2content)
```

```
## [1] 10 12 14 16
```

This code will display the content of row 2 from the multiplied matrix.

#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?

```
result <- matrixmul[1:2, 3:4]
print(result)
```

```
##      [,1] [,2]
## [1,]    6    8
## [2,]   14   16
```

This code extracts and displays the values from columns 3 and 4 for the first two rows of the matrix.

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

```
result <- matrixmul[3, 2:3]
print(result)
```

```
## [1] 24 26
```

This code extracts and displays the values from columns 2 and 3 for row 3 of the matrix.

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

```
result <- matrixmul[, 4]
print(result)
```

```
## [1]  8 16 28
```

This code extracts and displays all the values from column 4 of the matrix.

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

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

This code assigns the specified row and column names to the matrix.

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?

```
origmatrix <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4, byrow = TRUE)
reshapedmatrix <- matrix(origmatrix, nrow = 6, ncol = 2)
print(reshapedmatrix)
```

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

This code reshapes the original matrix to have 2 columns and 6 rows.

Using Arrays

- **Array can have more than two dimensions by using the array() function and dim() to specify the dimensions**

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?

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

```
## , , 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
```

This code creates a three-dimensional array with 4 columns and 2 rows, containing the specified values repeated twice.

b. How many dimensions do your array have?

```
dim(arrayval)
```

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

This output indicates that the array has 3 dimensions. The dimensions correspond to 2 rows, 4 columns, and 3 layers.

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?

```
arrayval <- array(rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), each = 2), dim = c(2, 4, 3))
rownames(arrayval) <- letters[1:2]
colnames(arrayval) <- LETTERS[1:4]
dimnames(arrayval) <- list(rownames = letters[1:2], colnames = LETTERS[1:4], layers = c("1st-Dimensional
print(arrayval)
```

```
## , , layers = 1st-Dimensional Array
##
##         colnames
## rownames A B C D
##        a 1 2 3 6
##        b 1 2 3 6
##
## , , layers = 2nd-Dimensional Array
##
##         colnames
## rownames A B C D
##        a 7 8 9 0
##        b 7 8 9 0
##
## , , layers = 3rd-Dimensional Array
##
##         colnames
## rownames A B C D
##        a 3 4 5 1
##        b 3 4 5 1
```

This code creates the array arrayval with the specified names for rows, columns, and layers.