Worksheet-3a in R

Riza Angelique Pelaez

2024-10-01

```
#Using Vectors

#1.LETTERS

#a. You need to produce a vector that contains the first 11 letters.
first11 <- LETTERS[1:11]
(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(1,26, by = 2)]

(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[c(1, 5, 9, 15, 21)]

(vowels)
```

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

```
#d. Produce a vector that contains the last 5 lowercase letters.
last5lower <- letters[22:26]

(last5lower)
```

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

```
#e. Produce a vector that contains letters between 15 to 24 letters in lowercase.
lower15_24 <- letters[15:24]

(lower15_24)
```

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

```
#2. Vector of Temperature.

#a. Character of cities
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. Vectors of Temperature
temp <- c(42, 39, 34, 34, 30, 27)

(temp)
```

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

```
#c. Data frame to combine the city and the temp
citytemp <- data.frame(City = city, Tempreture = temp)

(citytemp)
```

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

```
#d. rename the columns using the names() function

names(citytemp) <- c("City", "Tempreture")
(citytemp)
```

```
##              City Tempreture
## 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.
str(citytemp)
```

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

```
#outputs the structure of citytemp
```

```
#f. The content of row 3 and row 4.

(citytemp[3:4, ])
```

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

```
#g. Display the city with highest temperature and the city with the lowest temperature.
(citytemp[which.max(citytemp$Tempreture), ])
```

```
##              City Tempreture
## 1 Tuguegarao City         42
```

```r
(citytemp[which.min(citytemp$Tempreture), ])
```

```
##         City Tempreture
## 6 Davao City         27
```

```r
#Using Matrices

# 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
```

```r
## [,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
```

```r
## [,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
```

```r
## [,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
```

```
## [,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
```

```
# 3. Matrix with numbers from 1 to 8 and 11 to 14


# a. Create the matrix
mat <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)

(mat)
```

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

```
# b. Multiply the matrix by 2
matmult2 <- mat * 2

(matmult2)
```

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

```
# c. Display the content of row 2
(mat[2, ])
```

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

```
# d. Display columns 3 and 4 in row 1 and row 2
(mat[1:2, 3:4])
```

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

```
# e. Display columns 2 and 3 of row 3
(mat[3, 2:3])
```

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

```
# f. Display only column 4
(mat[, 4])
```

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

```
# g. Name the rows and columns
rownames(matmult2) <- c("isa", "dalawa", "tatlo")
colnames(matmult2) <- c("uno", "dos", "tres", "quatro")
(matmult2)
```

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

```
# h. Reshape the matrix with new dimensions (2 columns, 6 rows)
dim(mat) <- c(6, 2)
(mat)
```

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

```
#Using Array

# creates a two-dimensional array containing numbers from 1 to 24 that have 3 rows and 4 columns
arraydta <- array(c(1:24), c(3,4,2))
arraydta
```

```
## , , 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
```

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

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

```
#checking for the number of elements
length(arraydta)
```

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

```
vectorA <- c(1:24)
# creating an array
anArray <- array(vectorA, dim = c(3,4,2))
anArray
```

```
## , , 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
```

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

# a. Create an array with repeated values
arr <- array(rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), 2), dim = c(2, 4, 3))
(arr)
```

```
## , , 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. Check the dimensions of the array
(dim(arr))
```

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

```
# c. Name the rows, columns, and dimensions
dimnames(arr) <- list(letters[1:2], LETTERS[1:4], c("1st-Dimensional Array", "2nd-Dimensional Array", "3
(arr)
```

```
## , , 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
```