omisol

2023-10-10

#Number 1: #A.

```
first_11_letters <- LETTERS[1:11]
first_11_letters
```

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

#B.

```
odd_numbers <- seq(1, 26, by = 2)
odd_letters <- LETTERS[odd_numbers]
odd_letters
```

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

#C.

```
vowels <- LETTERS[LETTERS %in% c("A", "E", "I", "O", "U")]
vowels
```

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

#D.

```
last_5_low <- tail(letters, 5)
last_5_low
```

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

#E.

```
letters15to24 <- LETTERS[15:24]
letters15to24
```

```
##  [1] "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X"
```

#Number 2:

#A. The result is that city is a vector with specified city.

```
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. The result is that temp is a vector with specified temperatures

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

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

#C. The result is now a data frame which it combines the city and temp

```r
data <- data.frame(City = city, Temperature = temp)
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
```

#D. The column names now changed to City and Temperature

```r
names(data) <- c("City", "Temperature")
names(data)
```

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

#Number 3: #E The result the str()function will display the structure of datafram "data"

```r
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
```

#F it will retrieve the content of row 3 and row 4

```r
data[3, ]
```

```
##          City Temperature
## 3 Iloilo City          34
```

```r
data[4, ]
```

```
##       City Temperature
## 4 Tacloban          34
```

#G The result will Display that City with the highest and lowest temperature

```r
max_temp_city <- data[data$Temperature == max(data$Temperature), "City"]
min_temp_city <- data[data$Temperature == min(data$Temperature), "City"]
max_temp_city
```

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

```r
min_temp_city
```

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

USING MATRIX

#Number 2: #row = 2

```r
omi <-matrix(c(5,6,7,4,3,2,1,2,3,7,8,9),nrow = 2)
omi
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    5    7    3    1    3    8
## [2,]    6    4    2    2    7    9
```

```r
#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
matrix (data)
```

```
##      [,1]
## [1,] character,6
## [2,] numeric,6
```

#creating a diagonal matrix where value will always be 1

```r
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
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
```

#Number 2: #A.

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

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

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

#B.

```r
matrix(c(1:8, 11:14), nrow = 3, ncol = 4) * 2
```

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

```
#REsult is  value is multiplied by 2:
 #     [,1] [,2] [,3] [,4]
#[1,]    2    8   14   24
#[2,]    4   10   16   26
#[3,]    6   12   22   28
```

#C: REsult content of row to = 2 5 8 13

```
matrix_data <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
matrix_data[2, ]
```

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

#D

```
matrix_data <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
matrix_data[1:2, 3:4]
```

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

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

#E

```
matrix_data <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
matrix_data[3, 2:3]
```

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

#OUTPUT: 6 11 #F

```
matrix_data <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
matrix_data[, 4]
```

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

#OUTPUT: 12 13 14 #G

```
matrix_data <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
matrix_data
```

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

```
rownames(matrix_data) <- c("isa", "dalawa", "tatlo")
rownames(matrix_data)
```

```
## [1] "isa"    "dalawa" "tatlo"
```

```
colnames(matrix_data) <- c("uno", "dos", "tres", "quatro")
colnames(matrix_data)
```

```
## [1] "uno"    "dos"    "tres"   "quatro"
```

#H

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
matrix_data <- matrix(c(1:8, 11:14), nrow = 3, ncol = 4)
new_matrix <- matrix_data %>% t() %>% as.vector() %>% matrix(ncol = 2)
```

#USING ARRAYS

#Number 3: # B: result array has 3 dimensions: 2 rows, 4 columns, and 2 "layers" (depth). So, it is a three-dimensional array. # Given numeric values

```r
values <- c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1)
matrix_data <- matrix(rep(values, each = 2), nrow = 2)
array_data <- array(matrix_data, dim = c(2, 4, 2))
rownames(array_data) <- c("a", "b")
colnames(array_data) <- c("A", "B", "C", "D")
# Assign names to the dimensions
dimnames(array_data) <- list(
  "1st-Dimensional Array" = rownames(array_data),
  "2nd-Dimensional Array" = colnames(array_data),
  "3rd-Dimensional Array" = NULL
)
print(array_data)
```

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