

Rarg

CDS

2023-11-12

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##   Mean  :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##   Max.  :25.0    Max.    :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
#code here
```

```
#USING VECTORS
```

```
#1 There is a built-in vector LETTERS contains the uppercase letters of the alphabet and letters which
```

```
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"
```

```
#a. You need to produce a vector that contains the first 11 letters.
```

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

```
## [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(1,26,2)]  
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.
```

```
last_5_lowercase_letters <- tail(letters, 5)  
last_5_lowercase_letters
```

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

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

```
lowercase_letters <- letters[15:24]  
lowercase_letters
```

```
## [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
```

```
#a. What is the R code and its result for creating a character vector for the city/town of Tuguegarao
```

```
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 average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees. Name the object as temp
```

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

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

```
#c. Create a data frame to combine the city and the temp by using 'data.frame()'. What the R code and  
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. Associate the data frame you have created in 2.(c) by naming the columns using the names() functi  
names(data) <-c("City", "Temperature")  
names(data)
```

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

```
#f. From the answer in d, what is the content of row 3 and row 4 What is its R code and its output.  
row_3_and_4 <- data[3:4, ]  
row_3_and_4
```

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

```
#g. From the answer in d, display the city with highest temperature and the city with the lowest temp  
highest_temp_city <- data[data$Temperature == max(data$Temperature), "City" ]  
highest_temp_city
```

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

```
lowest_temp_city <- data[data$Temperature == min(data$Temperature), "City"]  
lowest_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_1 <- matrix(c(1:8, 11:14),3,4)  
matrix_1
```

```
##      [,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?  
matrix_Mul <- matrix_1 * 2
```

```

matrix_Mul

##      [,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?
row_2 <- matrix_1[2, ]
row_2

## [1]  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
Matrix_sub <- matrix_1[1:2, 3:4]
Matrix_sub

##      [,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_row_sub <- matrix_1[3, 2:3]
matrix_row_sub

## [1]  6 11

#f. What is the R code is you want to display only the columns 4? What is its output?
matrix_col4 <- matrix_1[, 4]
matrix_col4

## [1] 12 13 14

#g. Name the rows as isa, dalawa, tatlo and columns as uno, dos, tres, quatro for the matrix that was
rownames(matrix_1) <- c("isa", "dalawa", "tatlo")
colnames(matrix_1) <- c("uno", "dos", "tres", "quatro")

matrix_1

##      uno dos tres quatro
## isa      1  4    7     12
## dalawa   2  5    8     13
## tatlo    3  6   11     14

#h. From the original matrix you have created in a, reshape the matrix by assigning a new dimension w
reshape_matrix <- matrix_1
dim(reshape_matrix) <- c(6, 2)

reshape_matrix

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

```

#USING 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

```
num_values <-c( 1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1)
num_values
```

```
## [1] 1 2 3 6 7 8 9 0 3 4 5 1
```

```
array_1 <- array(num_values, dim = c(2,4,3))
array_1
```

```
## , , 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(array_1)
```

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

#c. Name the rows as lowercase letters and columns as uppercase letters starting from the A. The array

```
rownames(array_1) <- c("a","b")
```

```
array_1
```

```
## , , 1
```

```
##
```

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

```
## a      1    3    7    9
```

```
## b      2    6    8    0
```

```
##
```

```
## , , 2
```

```
##
```

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

```
## a      3    5    1    3
```

```
## b      4    1    2    6
```

```
##
```

```
## , , 3
```

```
##
```

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

```
## a      7    9    3    5
```

```
## b      8    0    4    1
```

```
colnames(array_1) <- c("A","B","C","D")
array_1
```

```
## , , 1
##
##   A B C D
## a 1 3 7 9
## b 2 6 8 0
##
```

```
## , , 2
##
##   A B C D
## a 3 5 1 3
## b 4 1 2 6
##
```

```
## , , 3
##
##   A B C D
## a 7 9 3 5
## b 8 0 4 1
```

```
dimnames(array_1)[[3]] <- c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional Array")
array_1
```

```
## , , 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
```