Worksheet-3a in R

RWorksheet_Sorenio#3

Worksheet for R Programming

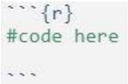
Instructions:

- Use RStudio or the posit(RStudio) Cloud accomplish this worksheet.

- Create folder for this worksheet#3. Inside the folder, create an .Rmd (R Markdown) for this worksheet and saved it as *RWorksheet_lastname#3a.Rmd*
- **Knit to pdf** to render a pdf file.
- On your own *GitHub repository*, push the .Rmd file, as well as the pdf worksheet knitted to the repo you have created before.
- Do not forget to comment your Git repo on our VLE
- Accomplish this worksheet by answering the questions being asked and writing the code manually.

Reminder

- To create a chunk of codes, you need to indicate this structure



~~~
```{r}
#code here

```
~~~

Figure 1: R Chunk

- You can add chunk options if you want, like:
  - echo: Whether to echo the source code in the output document (someone may not prefer reading your smart source code but only results).
    - Example: ```{r name, echo = TRUE}```

  - collapse: Whether to merge text output and source code into a single code block in the output. This is mostly cosmetic: collapse = TRUE makes the output more compact, since the R source code and its text output are displayed in a single output block. The default collapse = FALSE means R expressions and their text output are separated into different blocks

Chunk options in knitr are documented in (https://yihui.name/knitr/options) or you can see the R Mardown: The Definititive Guide for the whole document of R Markdown

You can check the Unit 3.4 notes posted in the VLE for the basic commands of R Markdown.

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
```

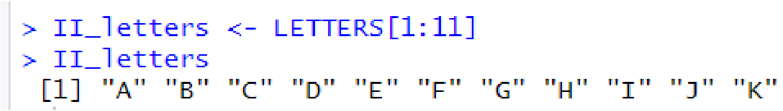
```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
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**:

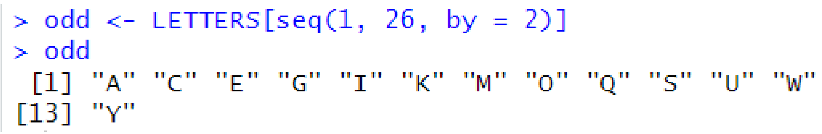
a. You need to produce a vector that contains the first 11 letters.



```
> II_letters <- LETTERS[1:11]
> II_letters
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K"
```

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

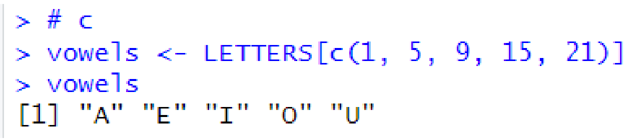
b. Produce a vector that contains the **odd numbered** letters.



```
> odd <- LETTERS[seq(1, 26, by = 2)]
> odd
 [1] "A" "C" "E" "G" "I" "K" "M" "O" "Q" "S" "U" "W"
[13] "Y"
```

odd <- LETTERS[seq(1, 26, by = 2)]
odd

c. Produce a vector that contains the **vowel**

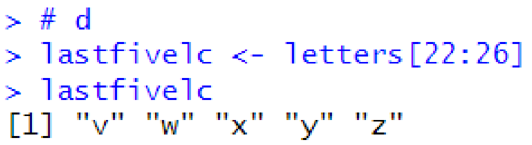


```
> # c
> vowels <- LETTERS[c(1, 5, 9, 15, 21)]
> vowels
[1] "A" "E" "I" "O" "U"
```

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

Based on the above vector **letters**:

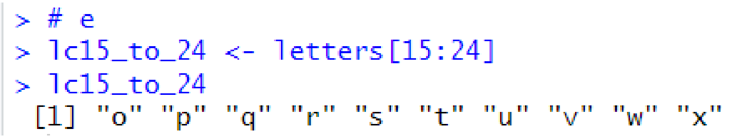
d. Produce a vector that contains the last 5 lowercase letters.



```
> # d
> lastfivelc <- letters[22:26]
> lastfivelc
[1] "v" "w" "x" "y" "z"
```

lastfivelc <- letters[22:26]
lastfivelc

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



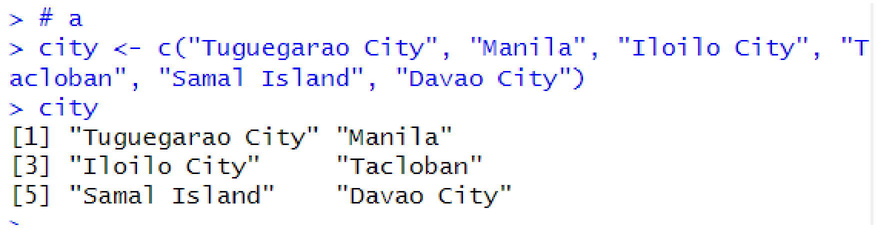
```
> # e
> lc15_to_24 <- letters[15:24]
> lc15_to_24
 [1] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
```

lc15_to_24 <- letters[15:24]
lc15_to_24

2. Create a *vector* (not a dataframe) with the average temperatures in April for Tugue-

garao City, Manila, Iloilo City, Tacloban, Samal Island, and Davao City. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees.

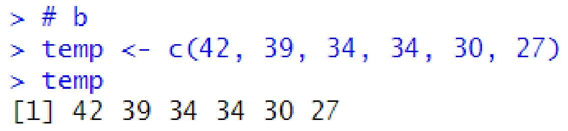
a. What is the R code and its result for creating a character vector for the city/town of Tuguegarao City, Manila, Iloilo City, Tacloban,Samal Island, and Davao City? Name the object as city. The names should follow the same order as in the instruction.



```
> # a
> city <- c("Tuguegarao City", "Manila", "Iloilo City", "T
acloban", "Samal Island", "Davao City")
> city
[1] "Tuguegarao City" "Manila"
[3] "Iloilo City"     "Tacloban"
[5] "Samal Island"    "Davao City"
```

city <- c("Tuguegarao City", "Manila", "Iloilo City", "Tacloban", "Samal Island", "Davao City")
city

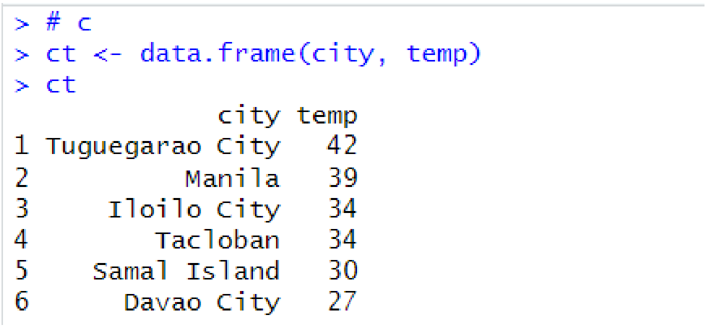
b. The average temperatures in Celcius are 42, 39, 34, 34, 30, and 27 degrees. Name the object as temp. Write the R code and its output. Numbers should also follow what is in the instruction.



```
> # b
> temp <- c(42, 39, 34, 34, 30, 27)
> temp
[1] 42 39 34 34 30 27
```

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

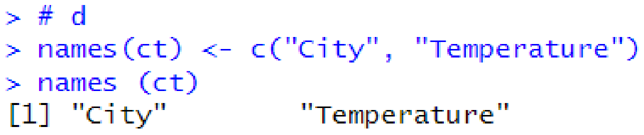
c. Create a dataframe to combine the city and the temp by using ‘data.frame(). What the R code and its result?



```
> # c
> ct <- data.frame(city, temp)
> ct
             city temp
1 Tuguegarao City   42
2          Manila   39
3     Iloilo City   34
4        Tacloban   34
5    Samal Island   30
6      Davao City   27
```

ct <- data.frame(city, temp)
ct

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?

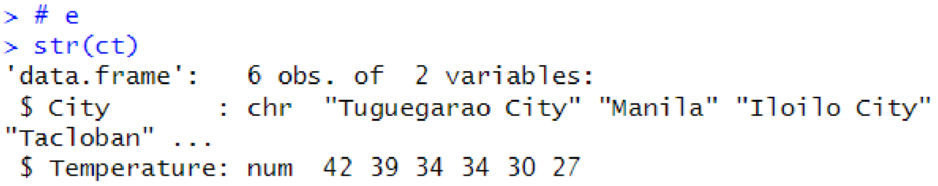


```
> # d
> names(ct) <- c("City", "Temperature")
> names (ct)
[1] "City"        "Temperature"
```

names(ct) <- c("City", "Temperature")

names (ct)

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

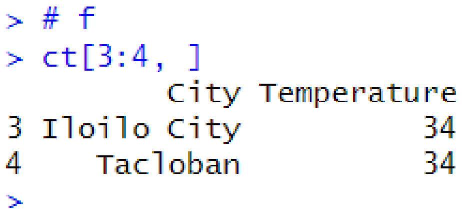


```
> # e
> str(ct)
'data.frame':	6 obs. of  2 variables:
 $ City       : chr  "Tuguegarao City" "Manila" "Iloilo City"
"Tacloban" ...
 $ Temperature: num  42 39 34 34 30 27
```

str(ct)

The dataframe ct combined the city and temp vectors using data.frame(). The dataframe organizes the city names and their corresponding average temperatures into two columns: "city" for the names and "temp" for the temperatures. This structure makes it easier to manage and analyze the data in a tabular format.

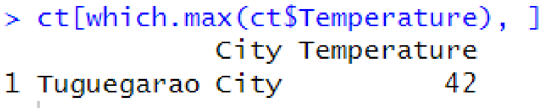
f. From the answer in d, what is the content of row 3 and row 4 What is its R code and its output?



```
> # f
> ct[3:4, ]
         City Temperature
3 Iloilo City          34
4    Tacloban          34
>
```

ct[3:4,]

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?



```
> ct[which.max(ct$Temperature), ]
             City Temperature
1 Tuguegarao City          42
```

```
ct[which.max(ct$Temperature), ]
```

Using Matrices

- Matrix can be created by specifying the rows and columns.

```
# 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
```

```
# 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
```

```
# creating a diagonal matrix where x value will always be 1
diag(1,nrow = 6,ncol = 5)
```

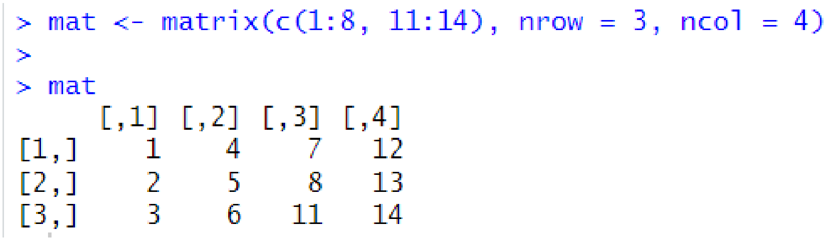
```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    1    0    0    0
## [3,]    0    0    1    0    0
## [4,]    0    0    0    1    0
## [5,]    0    0    0    0    1
## [6,]    0    0    0    0    0
```

```
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
```

2. Create a matrix of one to eight and eleven to fourteen with four columns and three rows.

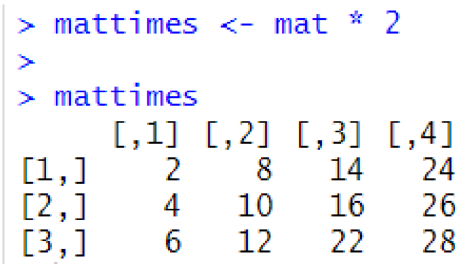
   a. What will be the R code for the #2 question and its result?



```
> 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
```

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

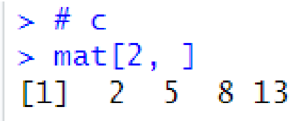
b. Multiply the matrix by two. What is its R code and its result?



```
> mattimes <- mat * 2
>
> mattimes
     [,1] [,2] [,3] [,4]
[1,]    2    8   14   24
[2,]    4   10   16   26
[3,]    6   12   22   28
```

mattimes <- mat * 2
mattimes

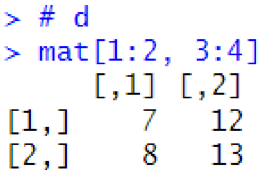
c. What is the content of row 2? What is its R code?



```
> # c
> mat[2, ]
[1]  2  5  8 13
```

mat[2,]

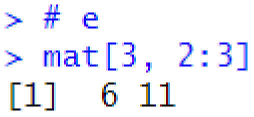
d. What will be the R code if you want to display the column 3 and column 4 in row 1 and row 2? What is its output?



```
> # d
> mat[1:2, 3:4]
     [,1] [,2]
[1,]    7   12
[2,]    8   13
```

mat[1:2, 3:4]

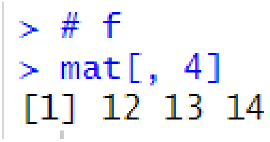
e. What is the R code is you want to display only the columns in 2 and 3, row 3? What is its output?



```
> # e
> mat[3, 2:3]
[1]  6 11
```

mat[3, 2:3]

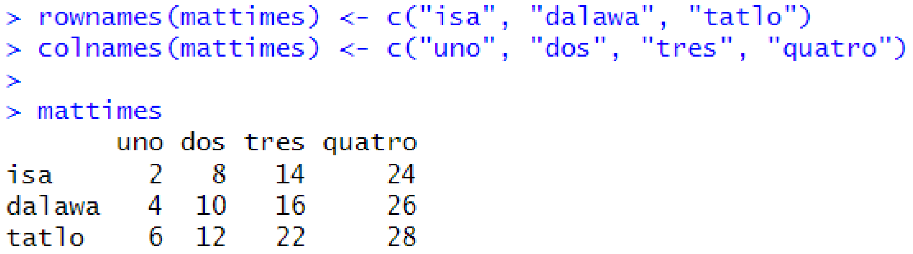
f. What is the R code is you want to display only the columns 4? What is its output?



```
> # f
> mat[, 4]
[1] 12 13 14
```

mat[, 4]

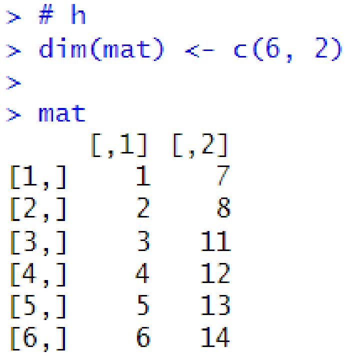
g. Name the rows as isa, dalawa, tatlo and columns as uno, dos, tres, quatro for the matrix that was created in **b.**‘. What is its R code and corresponding output?



```
> rownames(mattimes) <- c("isa", "dalawa", "tatlo")
> colnames(mattimes) <- c("uno", "dos", "tres", "quatro")
>
> mattimes
        uno dos tres quatro
isa       2   8   14     24
dalawa    4  10   16     26
tatlo     6  12   22     28
```

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

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?



```
> # h
> 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
```

dim(mat) <- c(6, 2)

Using Arrays

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

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

```
## , , 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(array_dta)
```

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

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

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

- Another way to create arrays

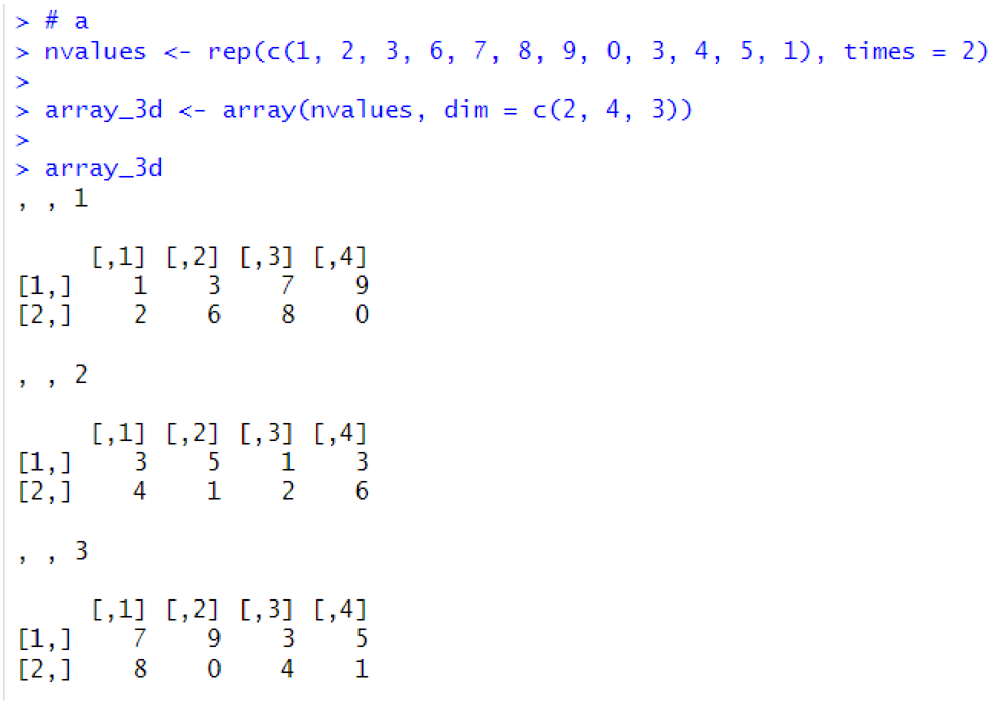
```
vectorA <- c(1:24)

# creating an array
an_Array <- array(vectorA, dim = c(3,4,2))
an_Array
```

```
## , , 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
```

2. 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

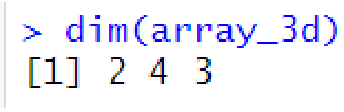
What will be the R code if you are to create a three-dimensional array with 4 columns and 2 rows. What will be its output?



```
> # a
> nvalues <- rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), times = 2)
>
> array_3d <- array(nvalues, dim = c(2, 4, 3))
>
> array_3d
, , 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
```

nvalues <- rep(c(1, 2, 3, 6, 7, 8, 9, 0, 3, 4, 5, 1), times = 2)
array_3d <- array(nvalues, dim = c(2, 4, 3))
array_3d

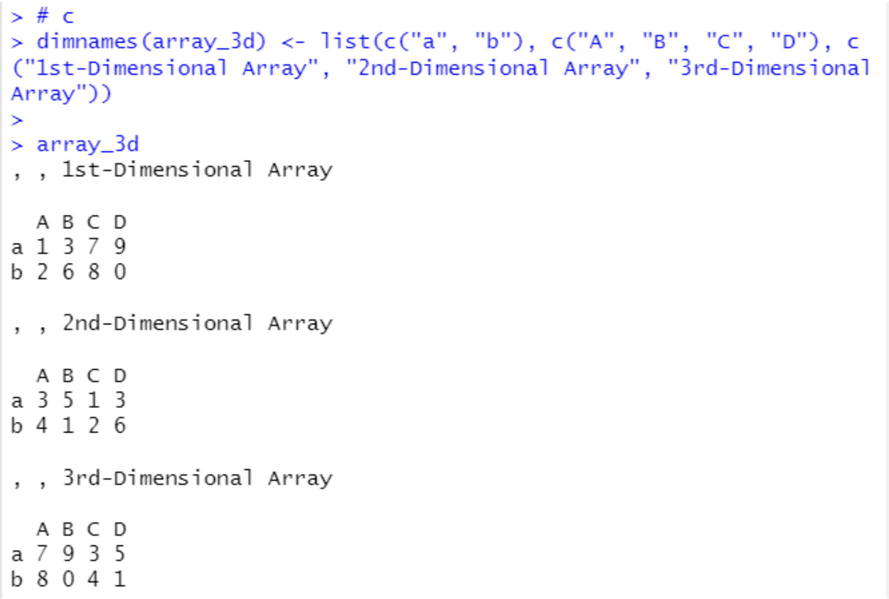
b. How many dimensions do your array have?



```
> dim(array_3d)
[1] 2 4 3
```

dim(array_3d)

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?



```
> # c
> dimnames(array_3d) <- list(c("a", "b"), c("A", "B", "C", "D"), c
("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional
Array"))
>
> array_3d
, , 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
```

dimnames(array_3d) <- list(c("a", "b"), c("A", "B", "C", "D"), c("1st-Dimensional Array", "2nd-Dimensional Array", "3rd-Dimensional Array"))

array_3d