

DATA ANALYTICS – 4027

LAB – 2

“CONTENTS”

- VECTOR
- MATRIX
- ARRAY
- LIST

Name: Hari Krishna P

Reg.Num: 19BCE7675

Date:23.10.21

Submitted To:

Prof. Hari Seetha

Vectors:

1. Create vector of numeric, complex, logical and character types of length 5.

```
> numbers <- c(1,2,3,4,5)
> numbers
[1] 1 2 3 4 5

> friends <- c("hari","krishna","christo","Paul","Harry")
> friends
[1] "hari"      "krishna" "christo" "Paul"     "Harry"

> complx <- c(3i+4,1i+2,2i+3,4i+1,3i-2)
> complx
[1] 4+3i 2+1i 3+2i 1+4i -2+3i

> log_values <- c(TRUE,FALSE,TRUE,FALSE,TRUE)
> log_values
```

2. Write a R program to add,multiply & divide two vectors of integers type and length 4

```
> #2
> numbers
[1] 1 2 3 4 5

> numbers1 = c(2,3,4,5,6)
> numbers
[1] 1 2 3 4 5
> numbers1
[1] 2 3 4 5 6

> nums <- numbers*numbers1
> nums
[1] 2 6 12 20 30
> nums <- numbers/numbers1
> nums
[1] 0.5000000 0.6666667 0.7500000 0.8000000 0.8333333
> nums <- numbers+numbers1
> nums
[1] 3 5 7 9 11
```

3. Write a R program to append value to a given empty vector

```
> #3
> vec = c()
> vec <- append(vec,"hello")
> vec
[1] "hello"
```

4. Write a R program to find Sum, Mean and Product of a Vector.

```
> #4
> vec1 <- c(1,2,3,4,5,6)
> vec1
[1] 1 2 3 4 5 6
> sum(vec1)
[1] 21
> mean(vec1)
[1] 3.5
> prod(vec1)
[1] 720
```

5. Write a R program to find Sum, Mean and Product of a Vector, ignore element like NA or NaN.

```
> #4
> vec1 <- c(1,2,3,4,5,6)
> vec1
[1] 1 2 3 4 5 6
> sum(vec1)
[1] 21
> mean(vec1)
[1] 3.5
> prod(vec1)
[1] 720
```

6. Write a R program to find the minimum and the maximum of a Vector

```
> #6
> min(vec1)
[1] 1
> max(vec1)
[1] 6
```

7. Write a R program to sort a Vector in ascending and descending order.

```

> vec1 <- c(2,3,56,46,221,1234,561,1)
> #ascending
> sort(vec1)
[1] 1 2 3 46 56 221 561 1234
> sort(vec1,decreasing = TRUE)
[1] 1234 561 221 56 46 3 2 1
>

```

8. Write a R program to test whether a given vector contains a specified element.

```

> #8
> x = c(11, 28, 2, 99, 7, 12)
> print(x)
[1] 11 28 2 99 7 12
> print(is.element(99, x))
[1] TRUE
> print(is.element(100, x))
[1] FALSE

```

9. Write a R program to find n^{th} highest value in a given vector

```

> x = c(11, 28, 2, 99, 7, 12)
> print(x)
[1] 11 28 2 99 7 12
> print(sort(x, TRUE)[n=1])
[1] 99

```

10. Write a R program to create a vector using : operator and seq() function

```

> #10
> num = seq(from=0,to=100,by=20)
> num
[1] 0 20 40 60 80 100
> #operator
> num =1:20
> num
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
> |

```

Lists:

1. Write a R program to create a list containing strings, numbers, vectors and a logical values

```
> myList <- list("Hari Krishna", c(1,2,3,4), TRUE,FALSE, 125.17)
> myList
[[1]]
[1] "Hari Krishna"

[[2]]
[1] 1 2 3 4

[[3]]
[1] TRUE

[[4]]
[1] FALSE

[[5]]
[1] 125.17
```

2. If Newlist <- list(a=1:10, b="Good morning", c="Hi"), write an R statement that will add 1 to each element of the first vector in Newlist.

```

> Newlist <- list(a=1:10, b="Namasthe", c="Hari")
>
> Newlist
$a
[1]  1  2  3  4  5  6  7  8  9 10

$b
[1] "Namasthe"

$c
[1] "Hari"

```

3. Consider `y <- list("a", "b", "c")`, write an R statement that will assign new names "one", "two" and "three" to the elements of y.

```

> x <- list("a", "b", "c")
> z <- list("one", "two", "three")
> append(x,z)
[[1]]
[1] "a"

[[2]]
[1] "b"

[[3]]
[1] "c"

[[4]]
[1] "one"

[[5]]
[1] "two"

[[6]]
[1] "three"

```

4. Let `string <- "Grand Opening"`, write an R statement to split this string into two and return the following output:

"Grand"
"Opening"

```
> string <- "Grand Opening"
> string.split <- strsplit(string, " ")
>
> mylist <- list(string.split[[1]][1], string.split[[1]][2])
> mylist
[[1]]
[1] "Hari Krishna"

[[2]]
[1] 1 2 3 4

[[3]]
[1] TRUE

[[4]]
[1] FALSE

[[5]]
[1] 125.17
```

5. Write a R program to select second element of a given nested list

```

> x <- list(list(0,2), list(3,4), list(5,6))
> print(x)
[[1]]
[[1]][[1]]
[1] 0

[[1]][[2]]
[1] 2

[[2]]
[[2]][[1]]
[1] 3

[[2]][[2]]
[1] 4

[[3]]
[[3]][[1]]
[1] 5

[[3]][[2]]
[1] 6

>
> y <- append(x, '[[', 2)
> print(y)
[[1]]
[[1]][[1]]
[1] 0

[[1]][[2]]
[1] 2

[[2]]
[[2]][[1]]
[1] 3

[[2]][[2]]
[1] 4

[[3]]
[1] "[["

[[4]]
[[4]][[1]]
[1] 5

[[4]][[2]]
[1] 6

```

6. Write a R program to merge two given lists into one list.


```

> ist1 <- list("Namasthe","India")
>
> ist2 <- list("Microsoft","Is Here!")
>
> list <- c(ist1,ist2)
> list
[[1]]
[1] "Namasthe"

[[2]]
[1] "India"

[[3]]
[1] "Microsoft"

[[4]]
[1] "Is Here!"

> |

```

7. Write a R program to convert a given list to vector.

```

> x <- list(1,2,3,4,5)
> x
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3

[[4]]
[1] 4

[[5]]
[1] 5

> y <- unlist(x)
>
> print(y)
[1] 1 2 3 4 5

```

8. Write a R program to add a new item a = "R Programming" to a given list

```

> Mylist <- list("Hello World")
>
> append(Mylist,"R programming",after=1)
[[1]]
[1] "Hello World"

[[2]]
[1] "R programming"

```

9. Write a R program to get the length of the first two vectors of a given list.

```

> Vector <- list(a<- c(1:5),b <-"Hari From", c<-"India" )
> Vector
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "Hari From"

[[3]]
[1] "India"

> length(Vector$a)
[1] 5
> length(Vector$b)
[1] 1
>
> length(Vector)
[1] 3

```

10. Write a R program to find all elements of a given list that are not in another given list.

```

> mylist1 <- list ("Hari","Vishesh","Christo","Ama1","Akilesh")
>
> myList2 <- list ("Jahnavi","Sam Stone","Hari","Christo")
>
> setdiff(mylist1,myList2)
[[1]]
[1] "Vishesh"

[[2]]
[1] "Ama1"

[[3]]
[1] "Akilesh"

```

Matrices:

1. Write a R program to create a matrix taking a given vector of numbers as input and define the column and row names. Display the matrix.

```

> friends <- matrix(c("Hari","Krishna","Janu","Harry","Christo","Vishesh"),nrow=3,ncol=3)
> friends
      [,1]      [,2]      [,3]
[1,] "Hari"    "Harry"  "Hari"
[2,] "Krishna" "Christo" "Krishna"
[3,] "Janu"    "Vishesh" "Janu"
> |

```

2. Write a R program to access the element at 3rd column and 2nd row, only the 3rd row and only the 4th column of a given matrix

```

> num = matrix(c(1,2,3,4,5,6,7,8,9,10,11,12),nrow = 4,ncol=4)
> num
      [,1] [,2] [,3] [,4]
[1,]     1     5     9     1
[2,]     2     6    10     2
[3,]     3     7    11     3
[4,]     4     8    12     4
> print(num[2,3])
[1] 10

> print(num[3,])
[1]  3  7 11  3

> print(num[,4])
[1] 1 2 3 4
> |

```

3. Write a R program to create two 2x3 matrix and add, subtract, multiply and divide the matrixes

```

> n1 = matrix(c(1,2,5,6,3,4), nrow = 2)
> n2 = matrix(c(7,8,9,10,11,12), nrow = 2)
> n1
      [,1] [,2] [,3]
[1,]     1     5     3
[2,]     2     6     4
> n2
      [,1] [,2] [,3]
[1,]     7     9    11
[2,]     8    10    12

```

```

> ans = n1 + n2
> ans
      [,1] [,2] [,3]
[1,]    8   14   14
[2,]   10   16   16
>
>
> ans = n1 - n2
> ans
      [,1] [,2] [,3]
[1,]   -6   -4   -8
[2,]   -6   -4   -8
>
>
> ans = n1 * n2
> ans
      [,1] [,2] [,3]
[1,]    7   45   33
[2,]   16   60   48
>
>
> ans = n1 / n2
> ans
      [,1]      [,2]      [,3]
[1,] 0.1428571 0.5555556 0.2727273
[2,] 0.2500000 0.6000000 0.3333333
> |

```

4. Write a R program to create a matrix from a list of given vectors

```

> for(x in friends){
+   print(x)
+ }
[1] "Hari"
[1] "Krishna"
[1] "Janu"
[1] "Harry"
[1] "Christo"
[1] "Vishesh"
[1] "Hari"
[1] "Krishna"
[1] "Janu"
> |

```

5. Write a R program to find row and column index of maximum and minimum value in a given matrix

```
> ans1 = which(matrix1 == max(matrix1), arr.ind=TRUE)
> ans1
      row col
[1,]    4    5
> ans1 = which(matrix1 == min(matrix1), arr.ind=TRUE)
> ans1
      row col
[1,]    1    1
> |
```

Arrays

6. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors

```
> v1 <- c(1,2,3,4,5)
> v1
[1] 1 2 3 4 5
> v2 <- c(15,16,17,18,19,20,21)
> v2
[1] 15 16 17 18 19 20 21
> ans <- array(c(v1,v2),dim = c(3,3,2))
> ans
, , 1
      [,1] [,2] [,3]
[1,]     1     4    16
[2,]     2     5    17
[3,]     3    15    18

, , 2
      [,1] [,2] [,3]
[1,]    19     1     4
[2,]    20     2     5
[3,]    21     3    15
```

7. Write a R program to create an 3 dimensional array of 24 elements using the dim() function.

```
> vec <- sample(1:5,24,replace = TRUE)
> dim(vec) <- c(3,2,4)
> dim(vec)
[1] 3 2 4
> vec
, , 1
      [,1] [,2]
[1,]     5     5
[2,]     3     2
[3,]     1     4
, , 2
      [,1] [,2]
[1,]     2     4
[2,]     5     2
[3,]     2     3
, , 3
      [,1] [,2]
[1,]     2     2
[2,]     5     2
[3,]     2     1
, , 4
      [,1] [,2]
[1,]     2     2
[2,]     3     2
[3,]     3     5
```

8. Write a R program to create an array of two 3x3 matrices each with 3 rows and 3 columns from two given two vectors.

```
> v1 = c(1,2,3,4)
>
> v2 = c(10,11,12,13,14,15)
> v1 <- c(1,2,3,4)
>
> v2 <- c(10,11,12,13,14,15)
> v1
[1] 1 2 3 4
> v2
[1] 10 11 12 13 14 15
> ans = array(c(v1,v2),dim = c(3,3,2))
> ans
, , 1
      [,1] [,2] [,3]
[1,]     1     4    12
[2,]     2    10    13
[3,]     3    11    14
, , 2
      [,1] [,2] [,3]
[1,]    15     3    11
[2,]     1     4    12
[3,]     2    10    13
```