

List of experiment

Unit-2

1. Write a R program to get the unique elements of a given string and unique numbers of vector

Code:

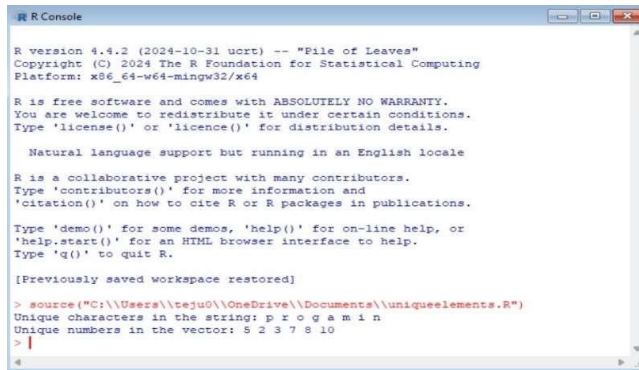
```
get_unique_chars <- function(input_string)
{
  unique_chars <- unique(strsplit(input_string,
  NULL)[[1]])  return(unique_chars)
}

get_unique_numbers <- function(input_vector) {
  unique_numbers <- unique(input_vector) # Extracting
  unique numbers from the vector  return(unique_numbers)
}

input_string <- "programming"
unique_chars <- get_unique_chars(input_string)
cat("Unique characters in the string:", unique_chars, "\n")

input_vector <- c(5, 2, 3, 3, 5, 7, 8, 8, 10)
unique_numbers <- get_unique_numbers(input_vector)
cat("Unique numbers in the vector:", unique_numbers,
"\n")
```

output:



```
R R Console

R version 4.4.2 (2024-10-31 ucrt) -- "File of Leaves"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> source("C:\\Users\\teju0\\OneDrive\\Documents\\uniqueselements.R")
Unique characters in the string: p r o g r a m i n
Unique numbers in the vector: 5 2 3 7 8 10
> |
```

2. Write a R program to create three vectors a,b,c with 3 integers. Combine the three vectors to become a 3×3 matrix where each column represents a vector. Print the content of the matrix.

Code:

```
a <- c(1, 4, 7)
```

```
b <- c(2, 5, 8)
```

```
c <- c(3, 6, 9)
```

```
matrix_combined <- cbind(a, b, c)
```

```
cat("The 3x3 matrix formed by combining vectors a, b, and  
c is:\n")
```

```
print(matrix_combined)
```

output:

```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\three vectors.R")
The 3x3 matrix formed by combining vectors a, b, and c is:
      a b c
[1,] 1 2 3
[2,] 4 5 6
[3,] 7 8 9
> |
```

3. Write a R program to create a list of random numbers in normal distribution on and count occurrences of each value.

Code:

```
random_numbers <- rnorm(100)
occurrences <- table(random_numbers)
print(occurrences)
```

output:

```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\random number.R")
random_numbers
-1.80394797419443 -0.676782109865056 -0.387697589471712 -0.154100225834817
      1              1              1              1
 0.190473111209614  0.232757187078284  0.340090192331851  0.385760582726031
      1              1              1              1
 0.517656992124836  1.24613607781684
      1              1
> |
```

4. Write a R program to read the .csv file and display the content

Code:

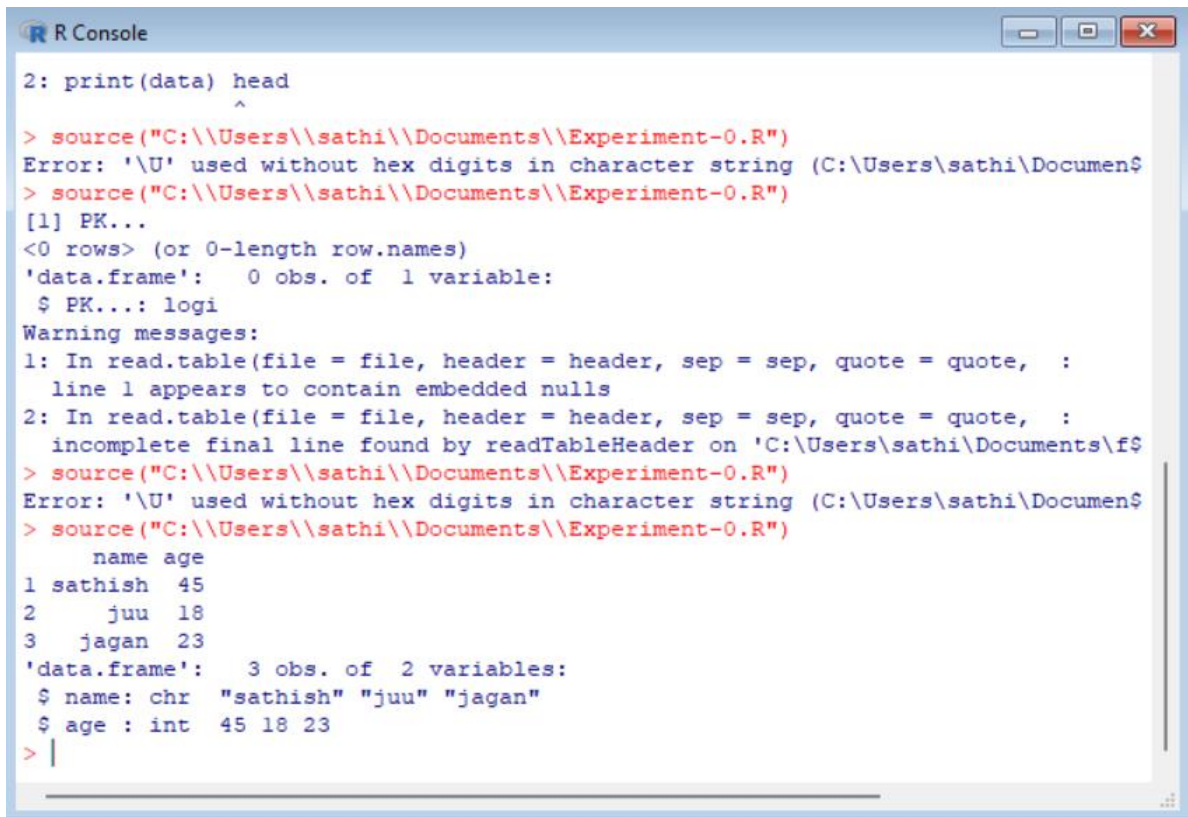
```
data <- read.csv("C:\\Users\\sathi\\Documents\\file2.csv")
```

```
print(data)
```

```
head(data)
```

```
str(data)
```

output:



```
R Console

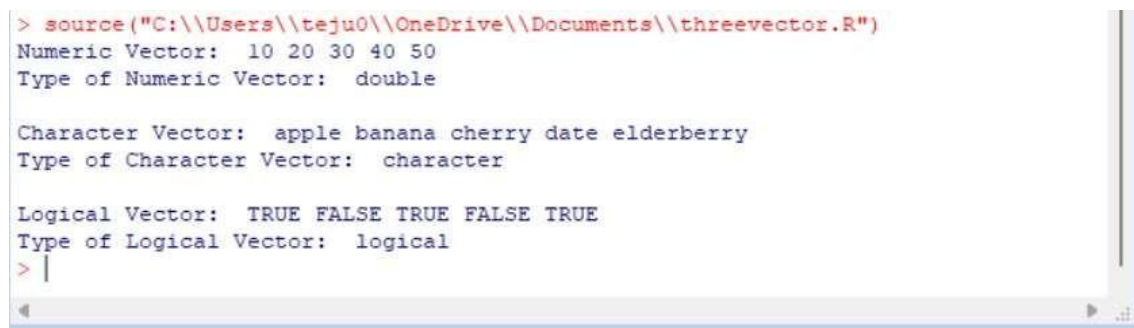
2: print(data) head
  ^
> source("C:\\Users\\sathi\\Documents\\Experiment-0.R")
Error: '\U' used without hex digits in character string (C:\Users\sathi\Documen$
> source("C:\\Users\\sathi\\Documents\\Experiment-0.R")
[1] PK...
<0 rows> (or 0-length row.names)
'data.frame': 0 obs. of 1 variable:
 $ PK...: logi
Warning messages:
1: In read.table(file = file, header = header, sep = sep, quote = quote, :
   line 1 appears to contain embedded nulls
2: In read.table(file = file, header = header, sep = sep, quote = quote, :
   incomplete final line found by readTableHeader on 'C:\Users\sathi\Documents\fs
> source("C:\\Users\\sathi\\Documents\\Experiment-0.R")
Error: '\U' used without hex digits in character string (C:\Users\sathi\Documen$
> source("C:\\Users\\sathi\\Documents\\Experiment-0.R")
      name age
1 sathish  45
2      juu  18
3      jagan 23
'data.frame': 3 obs. of 2 variables:
 $ name: chr  "sathish" "juu" "jagan"
 $ age : int  45 18 23
> |
```

5. Write a R program to create three vectors numeric data, character data and logical data. Display the content of the vectors and their type

Code:

```
numeric_vector <- c(10, 20, 30, 40, 50)
char_vector <- c("apple", "banana", "cherry", "date",
"elderberry")
logical_vector <- c(TRUE, FALSE, TRUE, FALSE, TRUE)
cat("Numeric Vector: ", numeric_vector, "\n") cat("Type of
Numeric Vector: ", typeof(numeric_vector),
"\n\n")
cat("Character Vector: ", char_vector, "\n")
cat("Type of Character Vector: ", typeof(char_vector), "\n\n")
cat("Logical Vector: ", logical_vector, "\n")
cat("Type of Logical Vector: ", typeof(logical_vector), "\n")
```

output:



```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\threevector.R")
Numeric Vector:  10 20 30 40 50
Type of Numeric Vector:  double

Character Vector:  apple banana cherry date elderberry
Type of Character Vector:  character

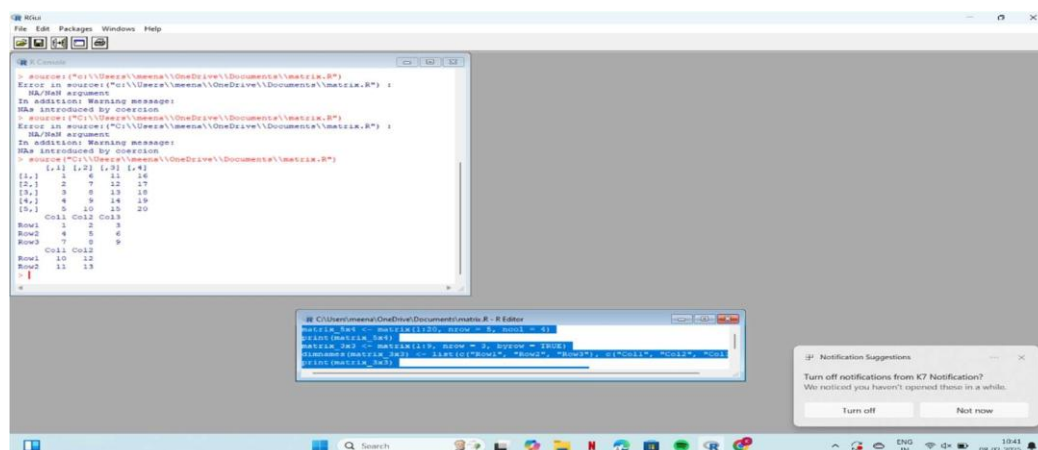
Logical Vector:  TRUE FALSE TRUE FALSE TRUE
Type of Logical Vector:  logical
> |
```

6. Write a R program to create a 5 x 4 matrix, 3 x 3 matrix with labels and fill the matrix by rows and 2 x 2 matrix with labels and fill the matrix by columns.

Code:

```
matrix_5x4 <- matrix(1:20, nrow = 5, ncol =  
4)  
print(matrix_5x4)  
matrix_3x3 <- matrix(1:9, nrow = 3, byrow = TRUE)  
dimnames(matrix_3x3) <- list(c("Row1", "Row2",  
"Row3"), c("Col1", "Col2", "Col3")) print(matrix_3x3)  
matrix_2x2 <- matrix(10:13, nrow = 2, byrow = FALSE)  
dimnames(matrix_2x2) <- list(c("Row1", "Row2"), c("Col1",  
"Col2"))  
print(matrix_2x2)
```

output:



The screenshot shows the RStudio interface. The console window displays the following output:

```
[1,] 1 2 3 4  
[2,] 5 6 7 8  
[3,] 9 10 11 12  
[4,] 13 14 15 16  
[5,] 17 18 19 20  
      Col1 Col2 Col3  
Row1  1  2  3  
Row2  5  6  7  
Row3  9 10 11  
      Col1 Col2  
Row1 10 12  
Row2 11 13
```

The script editor shows the code used to generate this output:


```
matrix_5x4 <- matrix(1:20, nrow = 5, ncol = 4)  
print(matrix_5x4)  
matrix_3x3 <- matrix(1:9, nrow = 3, byrow = TRUE)  
dimnames(matrix_3x3) <- list(c("Row1", "Row2", "Row3"), c("Col1", "Col2", "Col3"))  
print(matrix_3x3)  
matrix_2x2 <- matrix(10:13, nrow = 2, byrow = FALSE)  
dimnames(matrix_2x2) <- list(c("Row1", "Row2"), c("Col1", "Col2"))  
print(matrix_2x2)
```

7. Write a R program to create an array, passing in a vector of values and a vector of dimensions. Also, provide names for each dimension

Code:

```
my_array <- array(1:24, dim = c(2, 3, 4),  
dimnames = list(Row = c("Row1", "Row2"),  
                 Column = c("Col1", "Col2",  
                             "Col3"),  
                 Matrix = c("M1", "M2",  
                             "M3", "M4")) print(my_array)
```

output:




```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\array.R")  
The Array is:  
, , 1  
  
   [,1] [,2] [,3]  
[1,]    1    4    7  
[2,]    2    5    8  
[3,]    3    6    9  
  
, , 2  
  
   [,1] [,2] [,3]  
[1,]   10   13   16  
[2,]   11   14   17  
[3,]   12   15   18  
  
> |
```

8. Write an R program to create an array with three columns, three rows, and two "tables", taking two vectors as input to the array. Print the array.

Code:

```
vector1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9) vector2 <- c(10, 11,
12, 13, 14, 15, 16, 17, 18) array_data <-
array(c(vector1, vector2), dim = c(3, 3, 2)) cat("The
Array is: \n") print(array_data)
```

output:



```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\array.R")
The Array is:
, , 1
      [,1] [,2] [,3]
[1,]     1     4     7
[2,]     2     5     8
[3,]     3     6     9

, , 2
      [,1] [,2] [,3]
[1,]    10    13    16
[2,]    11    14    17
[3,]    12    15    18

> |
```


9. Write a R program to create a list of elements using vectors, matrices and a function. Print the content of the list

Code:

```
vec <- c(1, 2, 3) mat <-  
matrix(1:4, nrow = 2) add_fn  
<- function(x, y) x + y my_list  
<- list(vec, mat, add_fn)  
print(my_list)
```

output:



```
> source("C:\\Users\\teju0\\OneDrive\\Documents\\array.R")  
The Array is:  
, , 1  
  
      [,1] [,2] [,3]  
[1,]    1    4    7  
[2,]    2    5    8  
[3,]    3    6    9  
  
, , 2  
  
      [,1] [,2] [,3]  
[1,]   10   13   16  
[2,]   11   14   17  
[3,]   12   15   18  
  
> |
```

10. Write a R program to draw an empty plot and an empty plot specify the axes limits of the graphic

Code:

```
plot(1, type = "n")
```

```
plot(1, type = "n", xlim = c(0, 10), ylim = c(0, 20))
```

output:

