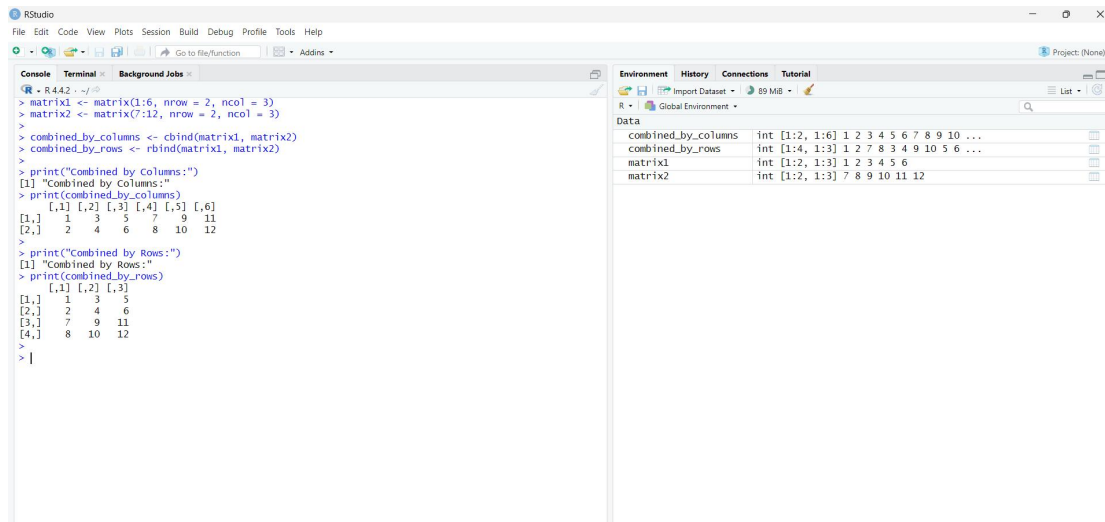


LAB--3

NAME : DUDEKULA MOHAMMAD ILYAS
REG NO :192325017

1. Perform the operation of combining matrices in R using cbind () and rbind() functions



The screenshot shows the RStudio interface. The console on the left contains the following R code and its output:

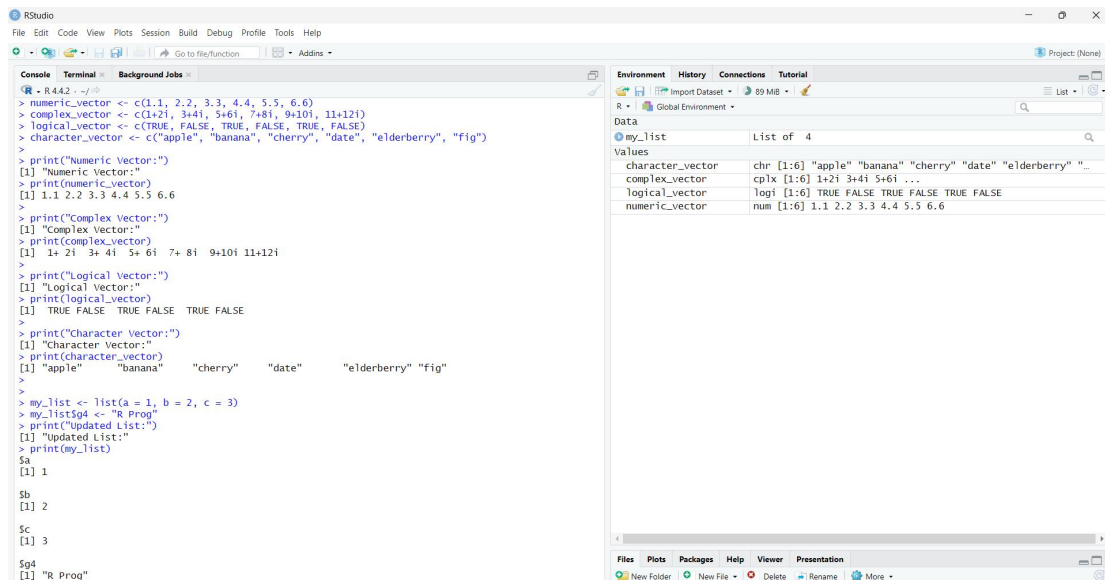
```
R - R442 - ~/R/
> matrix1 <- matrix(1:6, nrow = 2, ncol = 3)
> matrix2 <- matrix(7:12, nrow = 2, ncol = 3)
>
> combined_by_columns <- cbind(matrix1, matrix2)
> combined_by_rows <- rbind(matrix1, matrix2)
>
> print("Combined by Columns:")
[1] "Combined by Columns:"
> print(combined_by_columns)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    3    5    7    9   11
[2,]    2    4    6    8   10   12
>
> print("Combined by Rows:")
[1] "Combined by Rows:"
> print(combined_by_rows)
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
[3,]    7    9   11
[4,]    8   10   12
>
> |
```

The Environment pane on the right shows the objects created:

Object	Class	Attributes
combined_by_columns	int	[1:2, 1:6] 1 2 3 4 5 6 7 8 9 10 ...
combined_by_rows	int	[1:4, 1:3] 1 2 7 8 3 4 9 10 5 6 ...
matrix1	int	[1:2, 1:3] 1 2 3 4 5 6
matrix2	int	[1:2, 1:3] 7 8 9 10 11 12

2 .(i) Create vector of numeric, complex, logical and character with types of length 6.

(ii) Write a R program to add a new item g4 = "R Prog" to a given list.



The screenshot shows the RStudio interface. The console on the left contains the following R code and its output:

```
R - R442 - ~/R/
> numeric_vector <- c(1.1, 2.2, 3.3, 4.4, 5.5, 6.6)
> complex_vector <- c(1+2i, 3+4i, 5+6i, 7+8i, 9+10i, 11+12i)
> logical_vector <- c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE)
> character_vector <- c("apple", "banana", "cherry", "date", "elderberry", "fig")
>
> print("Numeric Vector:")
[1] "Numeric Vector:"
> print(numeric_vector)
[1] 1.1 2.2 3.3 4.4 5.5 6.6
>
> print("Complex Vector:")
[1] "Complex Vector:"
> print(complex_vector)
[1] 1+2i 3+4i 5+6i 7+8i 9+10i 11+12i
>
> print("Logical Vector:")
[1] "Logical Vector:"
> print(logical_vector)
[1] TRUE FALSE TRUE FALSE TRUE FALSE
>
> print("Character Vector:")
[1] "Character Vector:"
> print(character_vector)
[1] "apple" "banana" "cherry" "date" "elderberry" "fig"
>
> my_list <- list(a = 1, b = 2, c = 3)
> my_list$g4 <- "R Prog"
> print("Updated List:")
[1] "Updated List:"
> print(my_list)
$a
[1] 1

$b
[1] 2

$c
[1] 3

$g4
[1] "R Prog"
```

The Environment pane on the right shows the objects created:

Object	Class	Attributes
my_list	List	of 4
character_vector	chr	[1:6] "apple" "banana" "cherry" "date" "elderberry" ...
complex_vector	cp1x	[1:6] 1+2i 3+4i 5+6i ...
logical_vector	logi	[1:6] TRUE FALSE TRUE FALSE TRUE FALSE
numeric_vector	num	[1:6] 1.1 2.2 3.3 4.4 5.5 6.6

3. The price of one kg of rice is Rs. 40.75 and one kg of sugar is Rs. 30. Write R program to get the total amount of 2kg rice and 5kg sugar purchase

The screenshot shows the RStudio interface. The console on the left displays the following R code and its output:

```
R > R442 - ~/j/
> rice_price_per_kg <- 40.75
> sugar_price_per_kg <- 30
> rice_quantity <- 2
> sugar_quantity <- 5
> total_amount <- (rice_price_per_kg * rice_quantity) + (sugar_price_per_kg * sugar_quantity)
> print(paste("Total amount for 2kg rice and 5kg sugar: Rs.", total_amount))
[1] "Total amount for 2kg rice and 5kg sugar: Rs. 231.5"
>
> |
```

The Environment pane on the right shows the following variables and their values:

Variable	Value
rice_price_per_kg	40.75
rice_quantity	2
sugar_price_per_kg	30
sugar_quantity	5
total_amount	231.5

+

4. Write a R code to create a 3x4 matrix with 12 random numbers between 1-100; have the matrix be filled our row-by-row, instead of column-by-column. Name the columns of the matrix uno, dos, tres, cuatro, and the rows x, y, z. Scale the matrix by 10 and save the result

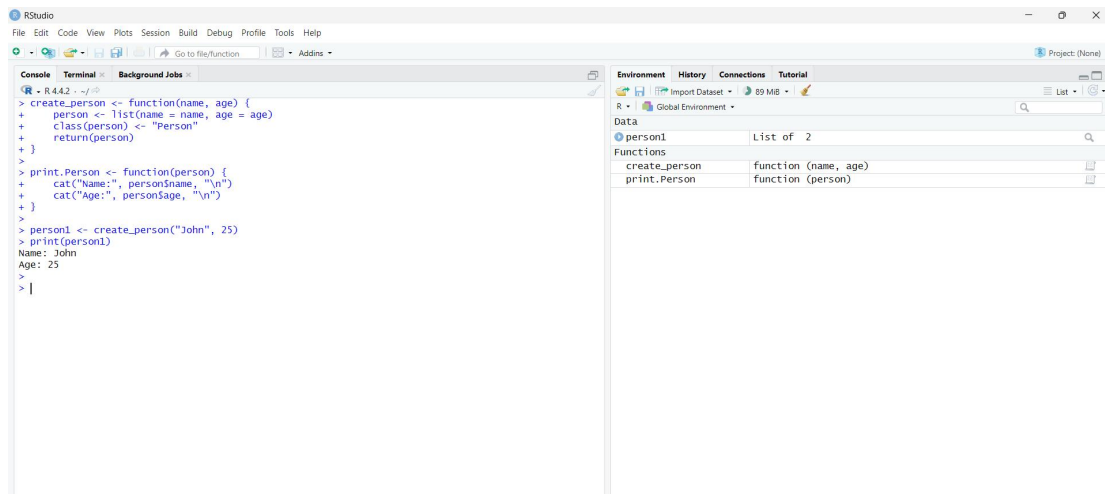
The screenshot shows the RStudio interface. The console on the left displays the following R code and its output:

```
R > R442 - ~/j/
> set.seed(123)
> matrix_data <- matrix(sample(1:100, 12, replace = TRUE), nrow = 3, ncol = 4, byrow = TRUE)
> colnames(matrix_data) <- c("uno", "dos", "tres", "cuatro")
> rownames(matrix_data) <- c("x", "y", "z")
> scaled_matrix <- matrix_data * 10
> print("Original Matrix:")
[1] "Original Matrix:"
> print(matrix_data)
      uno dos tres cuatro
x  31  79  51   14
y  67  42  50   43
z  14  25  90   91
> print("Scaled Matrix by 10:")
[1] "Scaled Matrix by 10:"
> print(scaled_matrix)
      uno dos tres cuatro
x 310 790 510 140
y 670 420 500 430
z 140 250 900 910
>
> |
```

The Environment pane on the right shows the following variables and their data types and dimensions:

Variable	Data Type	Dimensions	Values
matrix_data	int	[1:3, 1:4]	31 67 14 79 42 25 51 50 90 14 ...
scaled_matrix	num	[1:3, 1:4]	310 670 140 790 420 250 510 500 900 140 ...

5. Demonstrate the creation of S3 class in R.



The screenshot shows the RStudio interface with the following components:

- Console:** Displays the R code for creating an S3 class and an instance.

```
> create_person <- function(name, age) {  
+   person <- list(name = name, age = age)  
+   class(person) <- "Person"  
+   return(person)  
+ }  
>  
+ print.Person <- function(person) {  
+   cat("Name:", person$name, "\n")  
+   cat("Age:", person$age, "\n")  
+ }  
>  
+ person1 <- create_person("John", 25)  
+ print(person1)  
Name: John  
Age: 25  
>  
> |
```
- Environment:** Shows the global environment with a list named 'person1' containing 2 elements.
- Functions:** Lists the functions 'create_person' and 'print.Person'.

Data	
person1	List of 2

Functions	
create_person	function (name, age)
print.Person	function (person)