LAB

STATISTICS WITH R PROGRAMMING FOR VISUALIZATION

COURSE CODE: ITA0435

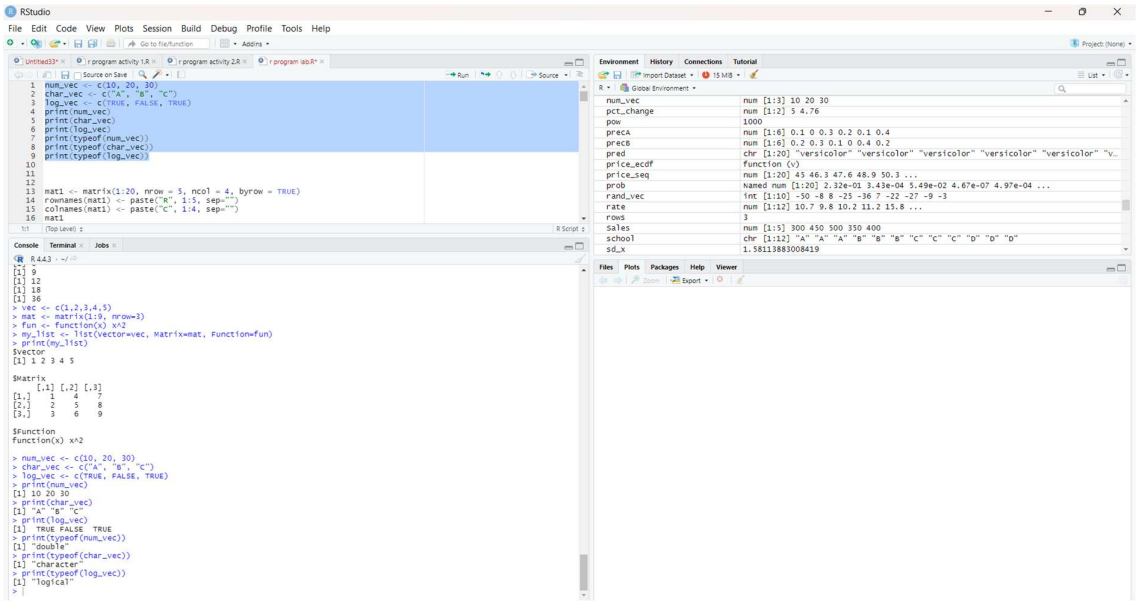
NAME:P.NIKITHA **REG NO:192324226**

1. **Create numeric, character, and logical vectors and display type and content.**

CODE:

```
num_vec <- c(10, 20, 30)

char_vec <- c("A", "B", "C")

log_vec <- c(TRUE, FALSE, TRUE)

print(num_vec)

print(char_vec)

print(log_vec)

print(typeof(num_vec))

print(typeof(char_vec))

print(typeof(log_vec))
```

OUTPUT:



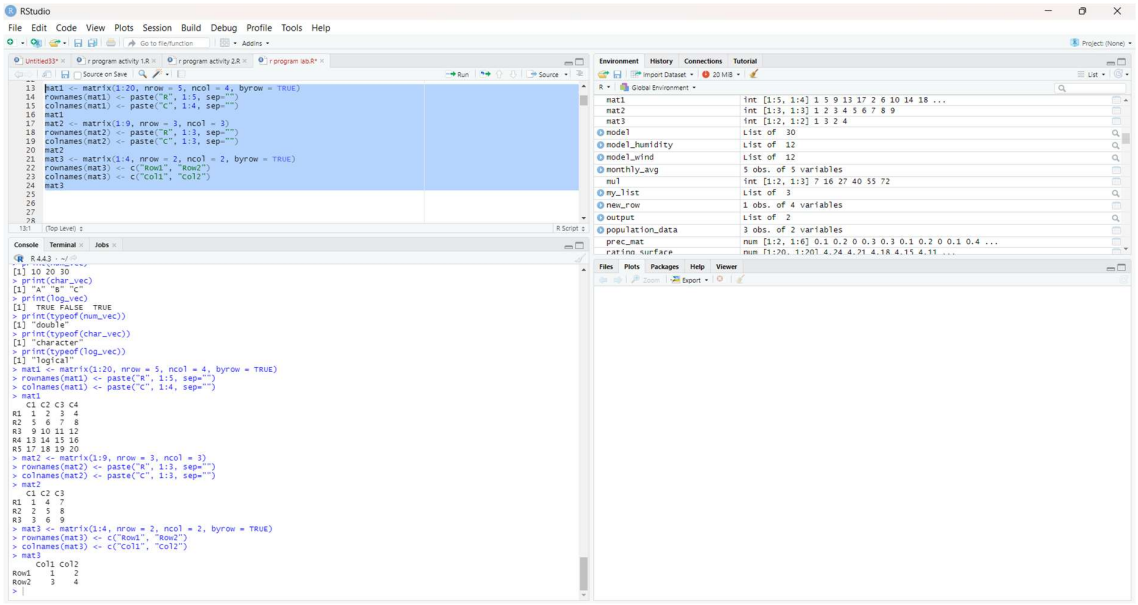
2. **Create labeled matrices (5×4, 3×3, 2×2) filled by row/column.**

CODE:

```
mat1 <- matrix(1:20, nrow = 5, ncol = 4, byrow = TRUE)

rownames(mat1) <- paste("R", 1:5, sep="")

colnames(mat1) <- paste("C", 1:4, sep="")
```

```
mat1
mat2 <- matrix(1:9, nrow = 3, ncol = 3)
rownames(mat2) <- paste("R", 1:3, sep="")
colnames(mat2) <- paste("C", 1:3, sep="")
mat2
mat3 <- matrix(1:4, nrow = 2, ncol = 2, byrow = TRUE)
rownames(mat3) <- c("Row1", "Row2")
colnames(mat3) <- c("Col1", "Col2")
mat3
```

OUTPUT:

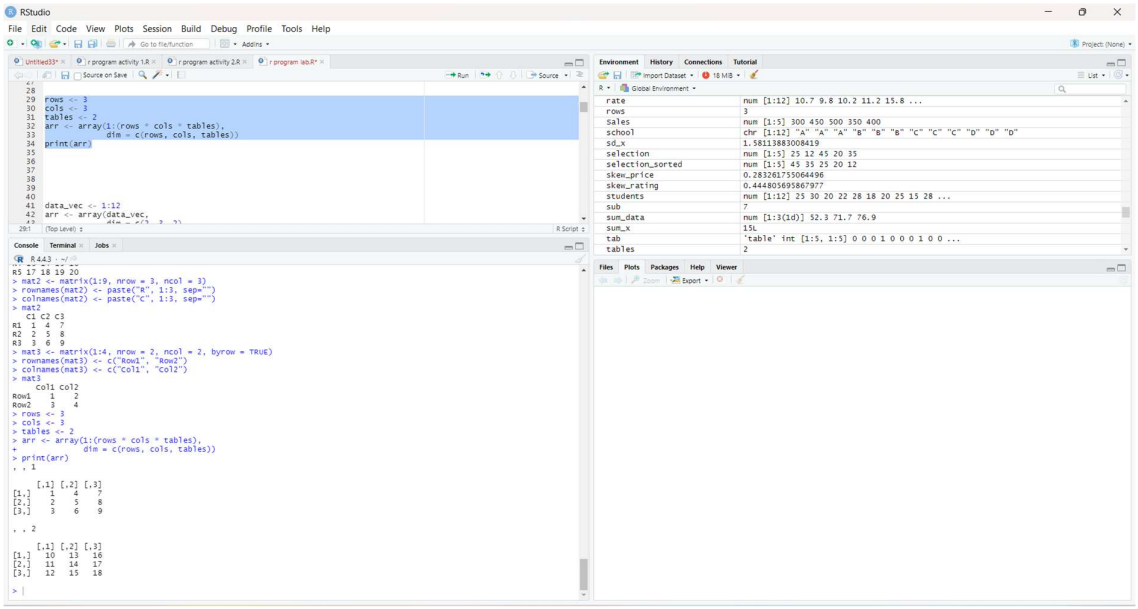


3. Write an R program to create and display a 3D array with specified rows, columns, and tables.

CODE:

```
rows <- 3
cols <- 3
tables <- 2
arr <- array(1:(rows * cols * tables),
             dim = c(rows, cols, tables))
print(arr)
```

OUTPUT:

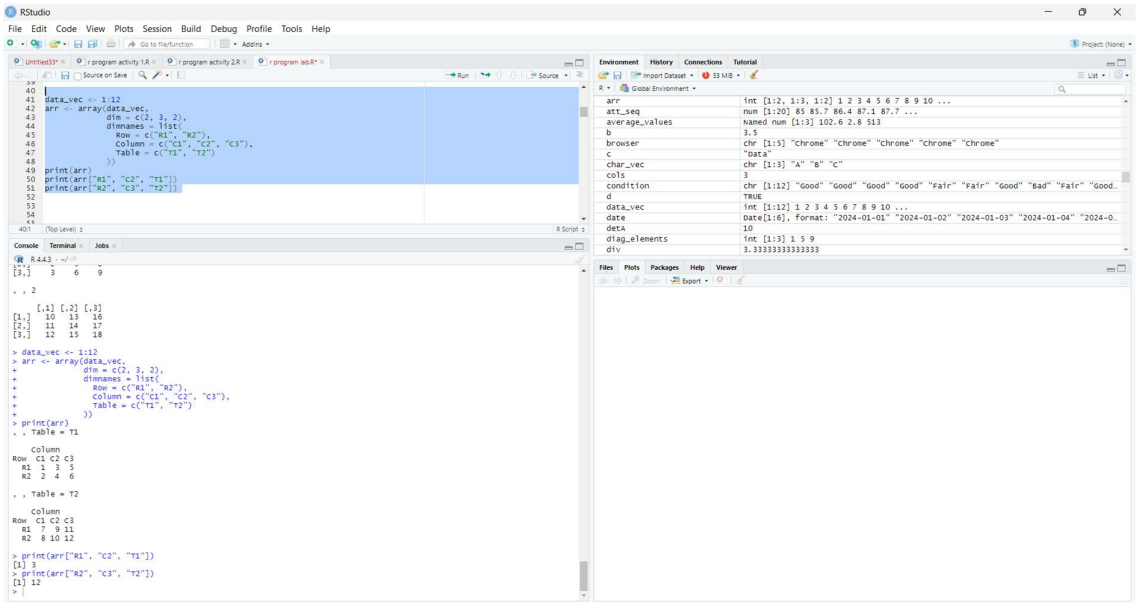


4. Create arrays from vectors with dimension names, print specific elements.

CODE:

```
data_vec <- 1:12

arr <- array(data_vec,

            dim = c(2, 3, 2),

            dimnames = list(

              Row = c("R1", "R2"),

              Column = c("C1", "C2", "C3"),

              Table = c("T1", "T2")

            ))

print(arr)

print(arr["R1", "C2", "T1"])

print(arr["R2", "C3", "T2"])
```

OUTPUT:

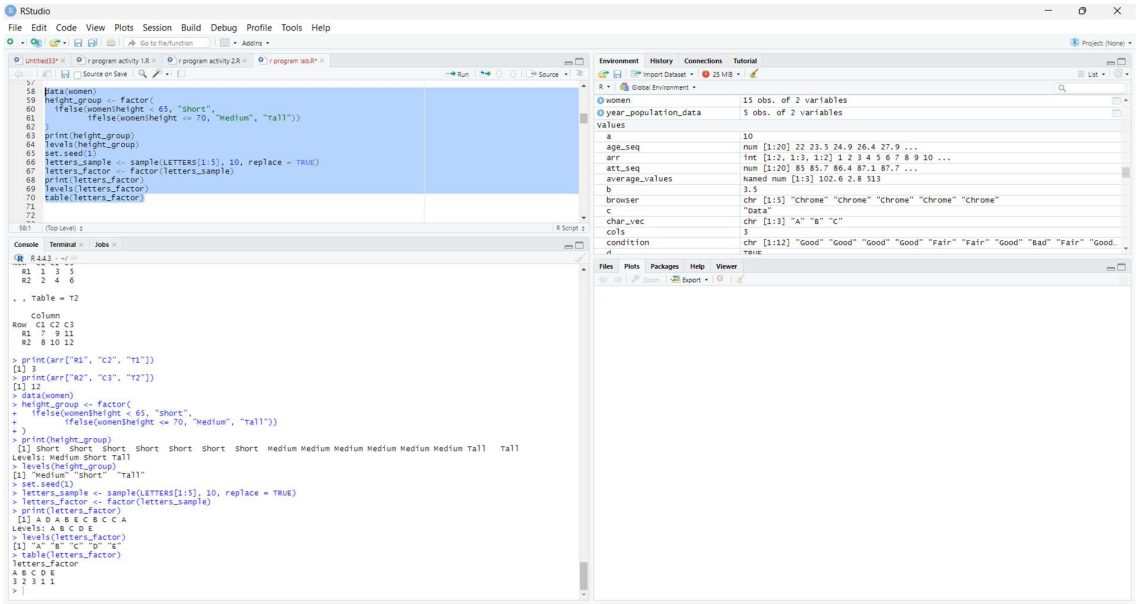


5. Create and manipulate factor variables (e.g., women's dataset heights, random LETTERS sample).

CODE:

```
data(women)

height_group <- factor(

  ifelse(women$height < 65, "Short",

         ifelse(women$height <= 70, "Medium", "Tall"))

)

print(height_group)

levels(height_group)

set.seed(1)

letters_sample <- sample(LETTERS[1:5], 10, replace = TRUE)

letters_factor <- factor(letters_sample)

print(letters_factor)

levels(letters_factor)

table(letters_factor)
```

OUTPUT:

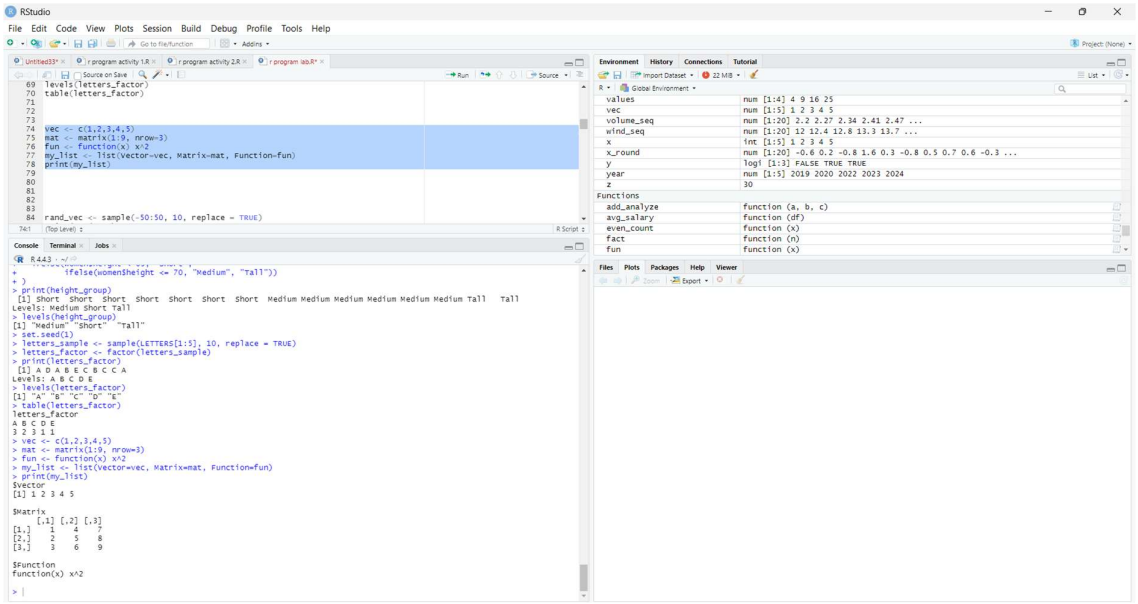


6. Create an R list containing vectors, matrices, and functions; display contents.

CODE:

```
vec <- c(1,2,3,4,5)

mat <- matrix(1:9, nrow=3)

fun <- function(x) x^2

my_list <- list(Vector=vec, Matrix=mat, Function=fun)

print(my_list)
```

OUTPUT:

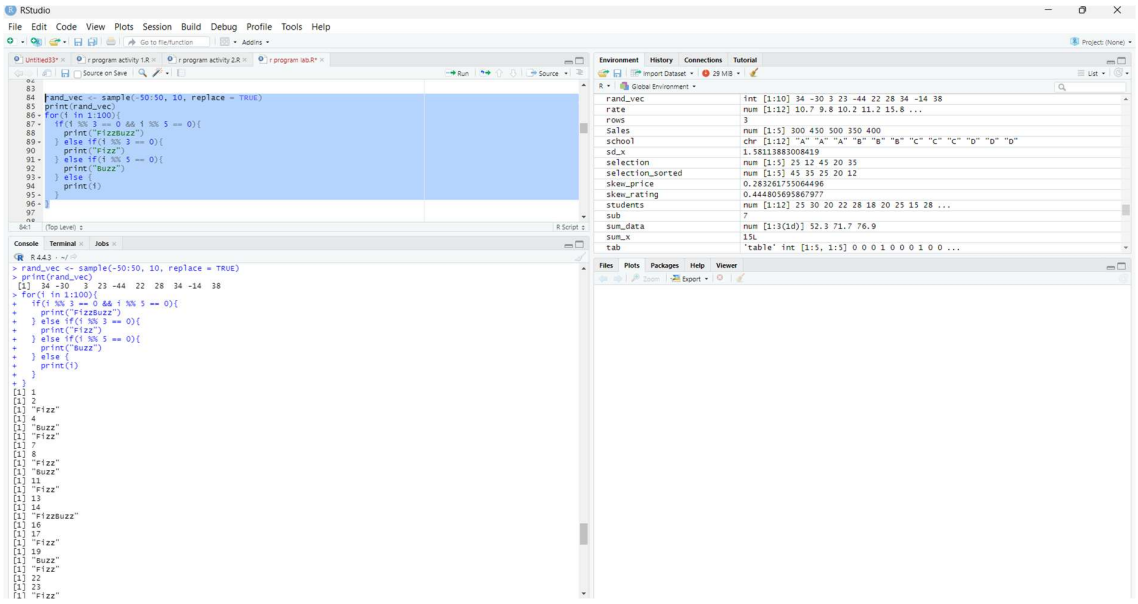


7. Write R programs for basic tasks: Factors of a number, generate a vector of 10 random integers between -50 and 50, print numbers 1–100 with FizzBuzz logic.

CODE:

```
rand_vec <- sample(-50:50, 10, replace = TRUE)
print(rand_vec)
for(i in 1:100){
  if(i %% 3 == 0 && i %% 5 == 0){
    print("FizzBuzz")
  } else if(i %% 3 == 0){
    print("Fizz")
  } else if(i %% 5 == 0){
    print("Buzz")
  } else {
    print(i)
  }
}
```

OUTPUT:



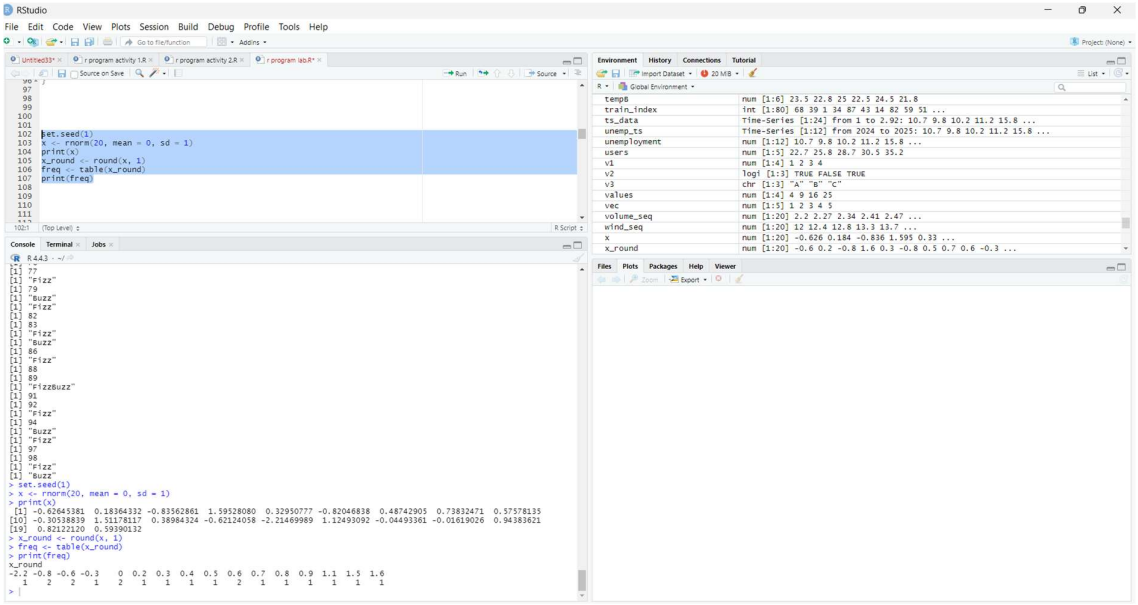
8. Generate random numbers from a normal distribution; count occurrences.

CODE:

```
set.seed(1)
x <- rnorm(20, mean = 0, sd = 1)
```

```
print(x)

x_round <- round(x, 1)

freq <- table(x_round)

print(freq)
```

OUTPUT:

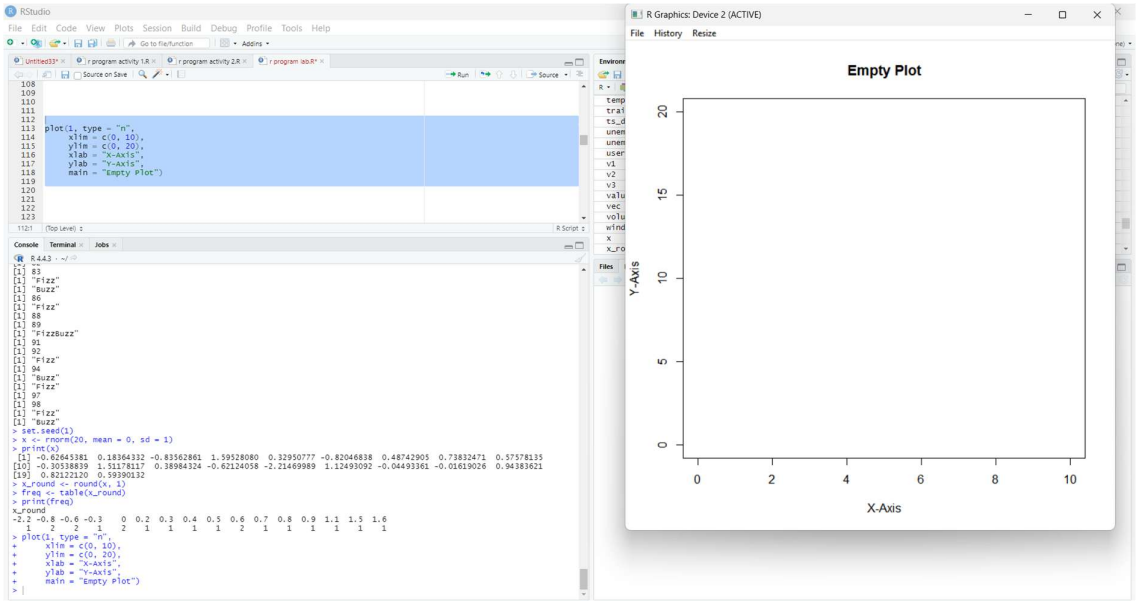


9. Create empty plots with specified axis limits.

CODE:

```
plot(1, type = "n",
     xlim = c(0, 10),
     ylim = c(0, 20),
     xlab = "X-Axis",
     ylab = "Y-Axis",
     main = "Empty Plot")
```

OUTPUT:



10. Create and explore a data frame exam_data with name, score, attempts, and qualify fields. Perform extract, add row/column, sort, save to file.

CODE:

```
exam_data <- data.frame(
  name = c("niki", "navya", "Chandu", "Divya"),
  score = c(85, 72, 90, 65),
  attempts = c(1, 2, 1, 3),
  qualify = c(TRUE, TRUE, TRUE, FALSE)
)
print(exam_data)
print(exam_data$score)
print(exam_data[1:2, ])
new_row <- data.frame(
  name = "Esha",
  score = 88,
  attempts = 1,
  qualify = TRUE
)
exam_data <- rbind(exam_data, new_row)
exam_data$grade <- c("B", "C", "A", "D", "B")
print(exam_data)
```

```
sorted_data <- exam_data[order(-exam_data$score), ]

print(sorted_data)

write.csv(exam_data, "exam_data.csv", row.names = FALSE)
```

OUTPUT: