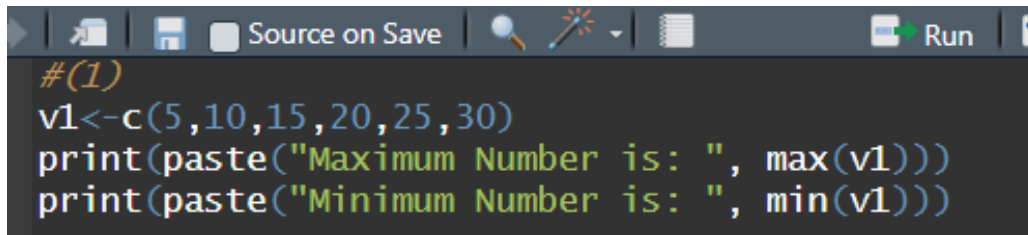


Probability and Statistics (UCS410)

Experiment 1: Basics of R programming

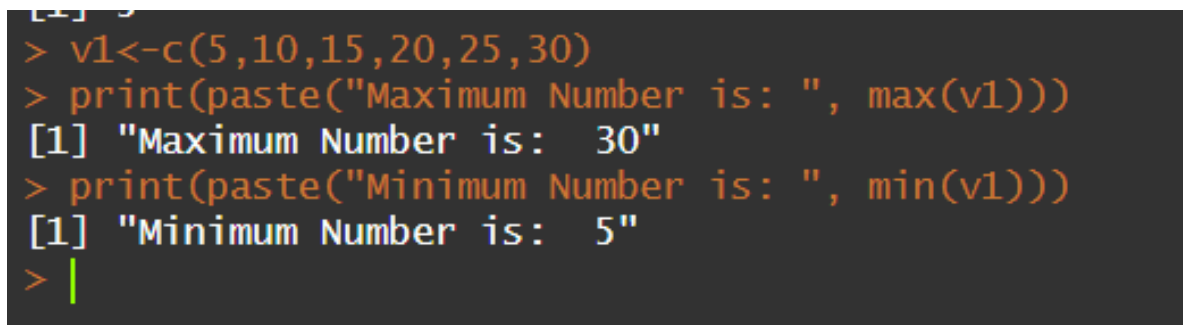
(1) Create a vector $c = [5, 10, 15, 20, 25, 30]$ and write a program which returns the maximum and minimum of this vector.

CODE:



```
#(1)
v1<-c(5,10,15,20,25,30)
print(paste("Maximum Number is: ", max(v1)))
print(paste("Minimum Number is: ", min(v1)))
```

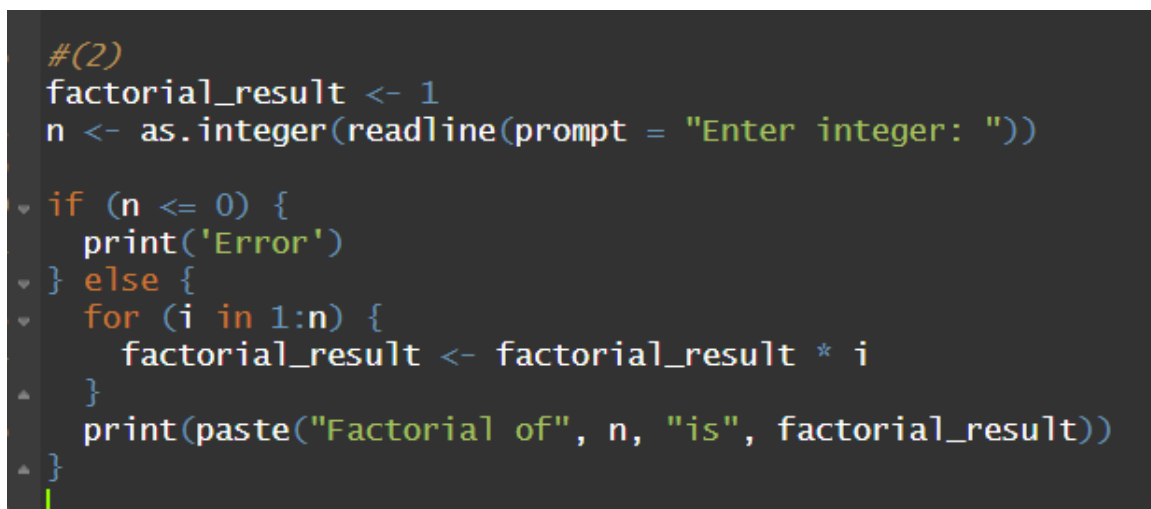
OUTPUT:



```
[1] 5
> v1<-c(5,10,15,20,25,30)
> print(paste("Maximum Number is: ", max(v1)))
[1] "Maximum Number is: 30"
> print(paste("Minimum Number is: ", min(v1)))
[1] "Minimum Number is: 5"
> |
```

(2) Write a program in R to find factorial of a number by taking input from user. Please print error message if the input number is negative.

CODE:



```
#(2)
factorial_result <- 1
n <- as.integer(readline(prompt = "Enter integer: "))

if (n <= 0) {
  print('Error')
} else {
  for (i in 1:n) {
    factorial_result <- factorial_result * i
  }
  print(paste("Factorial of", n, "is", factorial_result))
}
```

OUTPUT:

```
>
> #(2)
> factorial_result <- 1
> n <- as.integer(readline(prompt = "Enter integer: "))
Enter integer: 5
> if (n <= 0) {
+   print('Error')
+ } else {
+   for (i in 1:n) {
+     factorial_result <- factorial_result * i
+   }
+   print(paste("Factorial of", n, "is", factorial_result))
+ }
[1] "Factorial of 5 is 120"
> #(2)
> factorial_result <- 1
> n <- as.integer(readline(prompt = "Enter integer: "))
Enter integer: -1
> if (n <= 0) {
+   print('Error')
+ } else {
+   for (i in 1:n) {
+     factorial_result <- factorial_result * i
+   }
+   print(paste("Factorial of", n, "is", factorial_result))
+ }
[1] "Error"
> |
```

(3) Write a program to write first n terms of a Fibonacci sequence. You may take n as an input from the user.

CODE:

```
19
20 # (3)
21 n <- as.integer(readline("Enter the value of n: "))
22 if (n <= 2) {
23   print('Error')
24 } else {
25   fib <- numeric(n)
26   fib[1] <- 0
27   fib[2] <- 1
28
29   for (i in 3:n) {
30     fib[i] <- fib[i - 1] + fib[i - 2]
31   }
32
33   cat("Fibonacci sequence of", n, "terms:", fib)
34 }
35
```

OUTPUT:

```
> # (3)
> n <- as.integer(readline("Enter the value of n: "))
Enter the value of n: 1
> if (n <= 2) {
+   print('Error')
+ } else {
+   fib <- numeric(n)
+   fib[1] <- 0
+   fib[2] <- 1
+
+   for (i in 3:n) {
+     fib[i] <- fib[i - 1] + fib[i - 2]
+   }
+
+   cat("Fibonacci sequence of", n, "terms:", fib)
+ }
[1] "Error"
> # (3)
> n <- as.integer(readline("Enter the value of n: "))
Enter the value of n: 5
> if (n <= 2) {
+   print('Error')
+ } else {
+   fib <- numeric(n)
+   fib[1] <- 0
+   fib[2] <- 1
+
+   for (i in 3:n) {
+     fib[i] <- fib[i - 1] + fib[i - 2]
+   }
+
+   cat("Fibonacci sequence of", n, "terms:", fib)
+ }
Fibonacci sequence of 5 terms: 0 1 1 2 3
>
```

(4) Write an R program to make a simple calculator which can add, subtract, multiply and divide.

CODE:

```
#(4)
num1 <- as.numeric(readline("Enter the first number: "))
num2 <- as.numeric(readline("Enter the second number: "))

cat("Select operation:\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n")
choice <- as.integer(readline("Enter choice (1/2/3/4): "))

result <- switch(choice,
  "1" = num1 + num2,
  "2" = num1 - num2,
  "3" = num1 * num2,
  "4" = {
    if (num2 == 0) {
      stop("Error: Division by zero is not allowed.")
    }
    num1 / num2
  },
  stop("Error: Invalid choice."))

cat("Result:", result)
```

OUTPUT:

```
> #(4)
> num1 <- as.numeric(readline("Enter the first number: "))
Enter the first number: 5
> num2 <- as.numeric(readline("Enter the second number: "))
Enter the second number: 6
> cat("Select operation:\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n")
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
> choice <- as.integer(readline("Enter choice (1/2/3/4): "))
Enter choice (1/2/3/4): 3
> result <- switch(choice,
+   "1" = num1 + num2,
+   "2" = num1 - num2,
+   "3" = num1 * num2,
+   "4" = {
+     if (num2 == 0) {
+       stop("Error: Division by zero is not allowed.")
+     }
+     num1 / num2
+   },
+   stop("Error: Invalid choice."))
> cat("Result:", result)
Result: 30
> |
```

(5) Explore plot, pie, barplot etc. (the plotting options) which are built-in functions in R.

CODE:**PIE CHART:**

```
#(5)
# Load necessary library for plotting
install.packages("plotrix")
library(plotrix)

cities <- c("Kolkata", "Mumbai", "Delhi", "Chennai", "Patiala")
values <- c(5, 8, 30, 22, 55)

print(pie(values, labels = cities, main = "City Distribution"))
```

BAR GRAPH:

```
#BAR GRAPH
bar_colors <- c("red", "green", "blue", "orange", "purple")
barplot(values, names.arg = cities, col = bar_colors, main = "City Distribution")
```

HISTOGRAM:

```
#HISTOGRAM
data <- rnorm(100)
hist(data, main = "Histogram of Random Data", xlab = "Value", ylab = "Frequency", col = "blue")
```

SCATTER PLOT:

```
76 #SCATTER PLOT
77 x <- rnorm(50)
78 y <- 2 * x + rnorm(50)
79
80 plot(x, y, main = "Scatter Plot", xlab = "X", ylab = "Y", col = "red", pch = 19)
81
```

LINE PLOT:

```
#LINE PLOT
x <- seq(0, 2 * pi, length.out = 100)
y <- sin(x)

plot(x, y, type = "l", main = "Sine Function", xlab = "X", ylab = "Y", col = "green")
```

BOX PLOT:

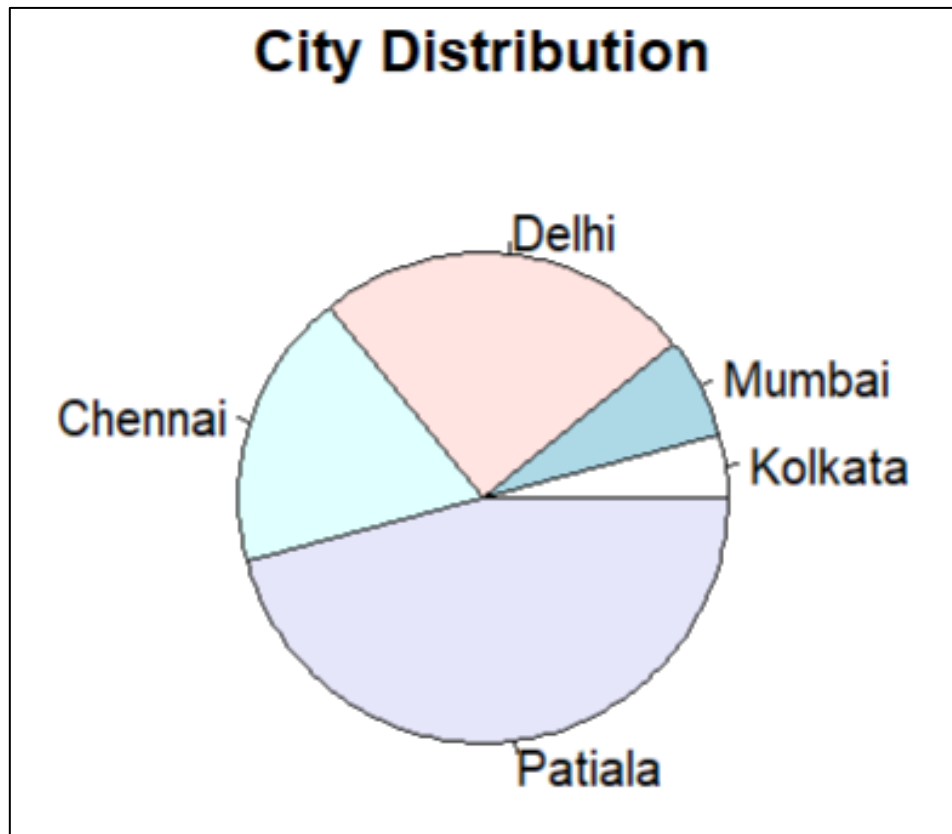
```
#BOX PLOT
data <- matrix(rnorm(200), ncol = 4) #random data
boxplot(data, main = "Box Plot of Random Data", col = c("red", "blue", "green", "purple"))
```

OUTPUT:**PIE CHART:**

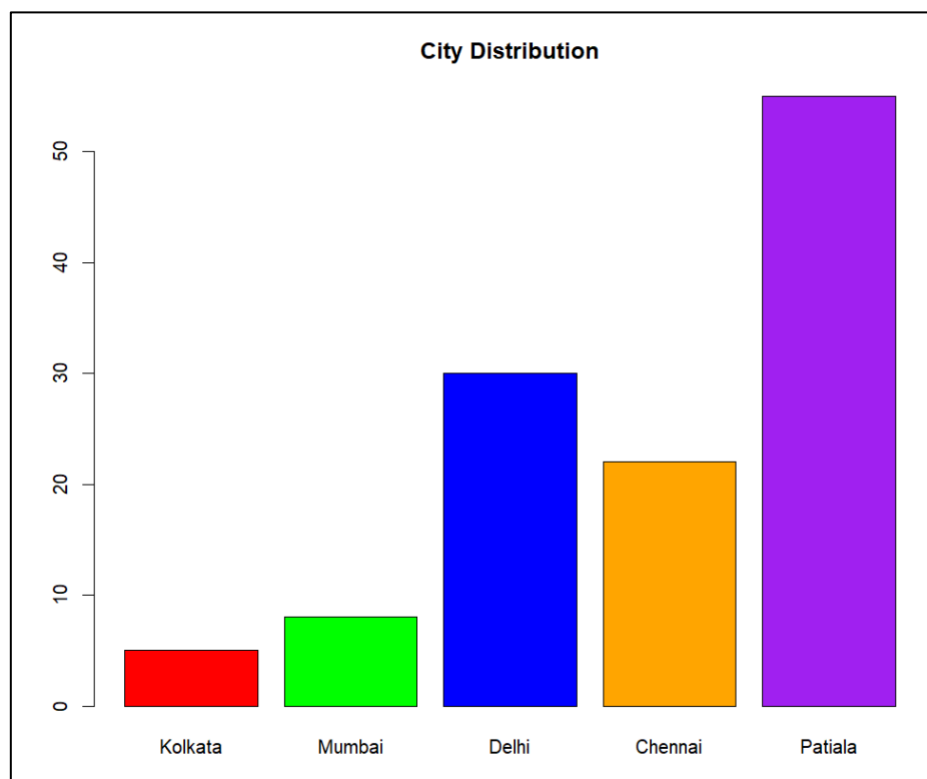
NAME: Shreeya Chatterji

ROLL NO: 102103447

CLASS: CO16



BAR GRAPH:

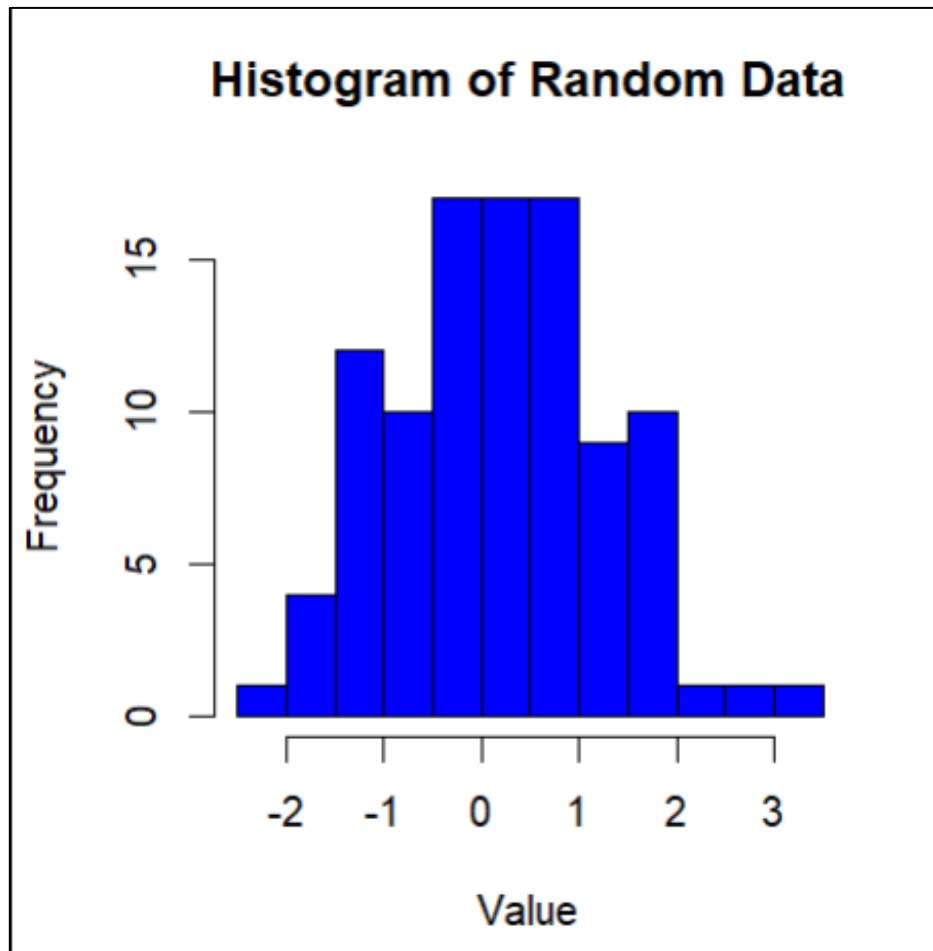


HISTOGRAM:

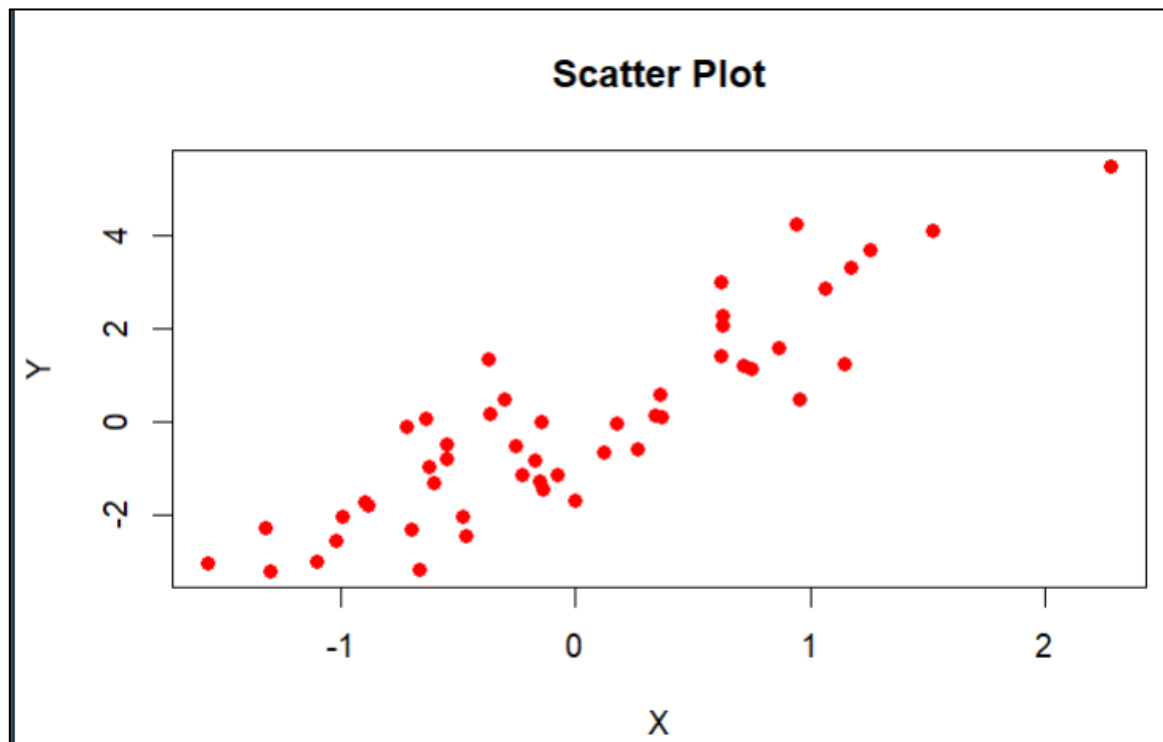
NAME: Shreeya Chatterji

ROLL NO: 102103447

CLASS: CO16



SCATTER PLOT:

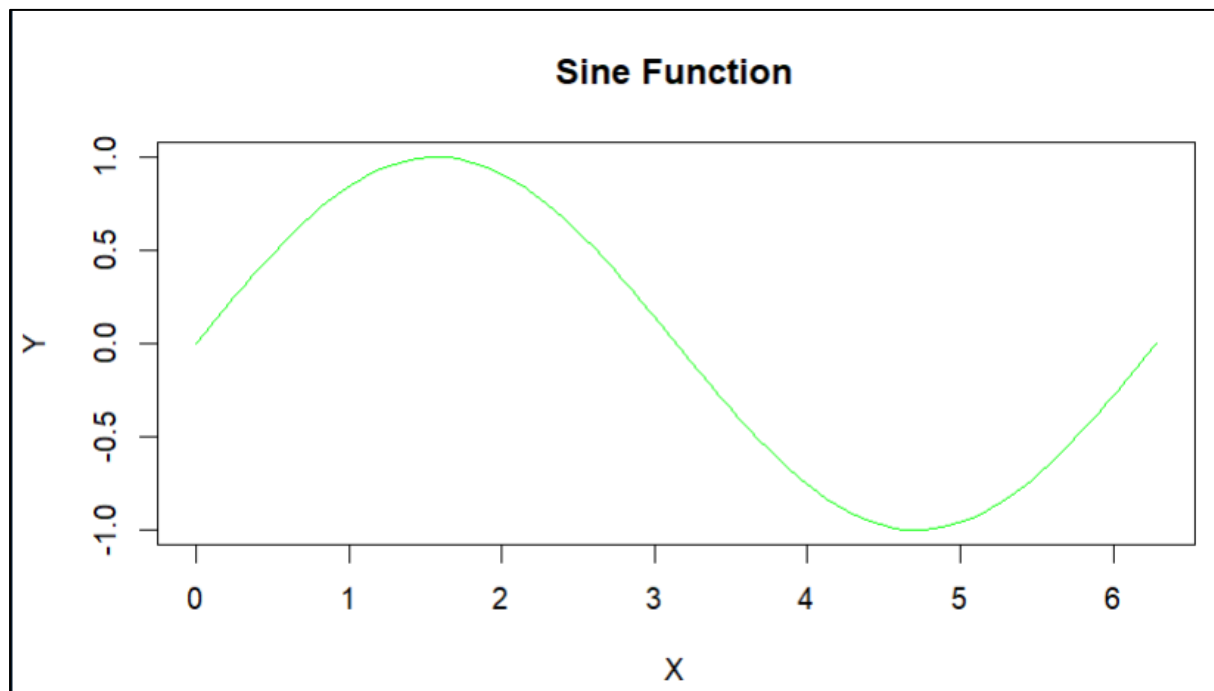


LINE PLOT:

NAME: Shreeya Chatterji

ROLL NO: 102103447

CLASS: CO16



BOX PLOT:

