Statistics Software Lab-Report : 1

Name: Shatansh Patnaik
Roll No: 20MA20067

List of Commands in the Script file:

```
# Basic Operations
1:10
(1:4)^2
(1:5)*10
((1:9)^2)[1]
(1:10)+(1:2)
factorial(10)
lfactorial(1)

# Plotting the graph
numLions <- function(x) choose(50, 5) * choose(x-50, 45)/choose(x, 50)
x <- 1:5000
plot(x, numLions(x), type="l")
result <- max(numLions(x))
result

# Cummulative Sums and Products
x <- c(1,2,3,4,5,6,7,8,9, 10)
cumsum(x)
cumprod(x)

# Matrix Operations
A <- matrix(c(1:10), nrow=5, byrow=TRUE)
print(A)

B <- matrix(c(1:10), nrow=5, byrow=FALSE)
print(B)

C <- matrix(c(1, 2, 3, 4, 5, 6), nrow=3, ncol=2, byrow = TRUE)
min(C[,1], C[,2])
pmin(C[1,], C[2,])

# NLM Function
nlm(function(x) return(x^2-9), 1)

# Calculus
D(expression(exp(x^2)), "x")
integrate(function(x) x^2, 0, 1)
```

```
# Sorting
a <- c(4,1,2,5,2)
y = sort(a)
y
order(a)

# Linear Algebra Ops
y <- c(1, 3, 5, 7)
2*y
crossprod(c(1,1,1), c(2,2,2)) # Actually dot

# Matrix Multiplication
H <- matrix(c(1,2,2,1), nrow=2, ncol=2)
H

H%*%H

# solutions of Equations
A <- matrix(c(1,2,1,1), nrow=2, ncol=2)
b <- c(1,2)
solve(A,b)
solve(A)

# Set Ops
x <- c(1,2,3,4)
y <- c(3,4,5,6)
union(x,y)
intersect(x,y)
2%in%y
choose (4,2)
```

CLASS PROBLEMS

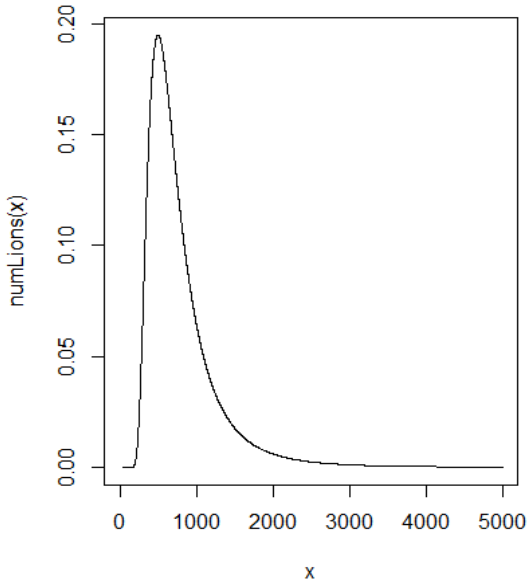
```
# Output should be 1, 2, 3 ... 15
for(i in 1:15)
{
  print(i)
}

# Output should be 1, 3, 6 ... 120
j=0
for(i in 1:15)
{
  j=j+i
  print(j)
}
```

```
# While Loop
# Output should be 1, 2, 3 ... 15
i=0
while(i<15){
  i=i+1
  print(i)
}

# Output should be 1, 3, 6 ... 120
i=0
j=0
while(i<15){
  i=i+1
  j=j+i
  print(j)
}

# Repeat Loop
# Output should be 1, 2, 3 ... 15
i=1
repeat {
  print(i)
  i=i+1
  if(i>15){
    break
  }
}
# Output should be 1, 3, 6 ... 120
i=1
j=1
repeat{
  print(j)
  i=i+1
  j=j+i
  if(i>15){
    break
  }
}

# Take two numbers as input and add, subtract, multiply, divide
{
a = readline("Enter A: ");
b = readline("Enter B: ");
}
a = as.numeric(a);
```

```
b = as.numeric(b);

# Standard Arithmetic Ops
print(a+b);
print(a-b);
print(a*b);
print(a/b);

# Matrix through Keyboard and then their multiplication
buildMatrix <- function() {
  ro <- as.numeric(readline("Enter the number of rows: "))
  cat("Enter the all the values row-wise: ")
  val <- scan(stdin())
  matrix(val, ro, byrow=TRUE)
}

A <- buildMatrix()
B <- buildMatrix()

print(A+B)
print(A-B)
print(A%*%B)
```

Output

```
> # Basic Operations
> 1:10
 [1]  1  2  3  4  5  6  7  8  9 10
> (1:4)^2
[1]  1  4  9 16
> (1:5)*10
[1] 10 20 30 40 50
> ((1:9)^2)[1]
[1] 1
> (1:10)+(1:2)
 [1]  2  4  4  6  6  8  8 10 10 12
> factorial(10)
[1] 3628800
> lfactorial(1)
[1] 0
```




```
> # Plotting the graph
> numLions <- function(x) choose(50, 5) * choose(x-50, 45)/choose(x, 50)
> x <- 1:5000
> plot(x, numLions(x), type="l")
> result <- max(numLions(x))
> result
[1] 0.1948912
>
> # Cummulative Sums and Products
> x <- c(1,2,3,4,5,6,7,8,9, 10)
> cumsum(x)
 [1]  1  3  6 10 15 21 28 36 45 55
> cumprod(x)
 [1]       1       2       6      24     120     720
 [7]    5040   40320  362880 3628800
>
> # Matrix Operations
> A <- matrix(c(1:10), nrow=5, byrow=TRUE)
> print(A)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
[5,]    9   10
>
> B <- matrix(c(1:10), nrow=5, byrow=FALSE)
> print(B)
     [,1] [,2]
[1,]    1    6
[2,]    2    7
```

```
[3,]    3    8
[4,]    4    9
[5,]    5   10
>
> C <- matrix(c(1, 2, 3, 4, 5, 6), nrow=3, ncol=2, byrow = TRUE)
> min(C[,1], C[,2])
[1] 1
> pmin(C[1,], C[2,])
[1] 1 2
>
> # NLM Function
> nlm(function(x) return(x^2-9), 1)
$minimum
[1] -9

$estimate
[1] -2.500222e-13

$gradient
[1] 1.000089e-06

$code
[1] 1

$iterations
[1] 1

>
> # Calculus
> D(expression(exp(x^2)), "x")
exp(x^2) * (2 * x)
> integrate(function(x) x^2, 0, 1)
0.3333333 with absolute error < 3.7e-15
>
> # Sorting
> a <- c(4,1,2,5,2)
> y = sort(a)
> y
[1] 1 2 2 4 5
> order(a)
[1] 2 3 5 1 4
>
> # Linear Algebra Ops
> y <- c(1, 3, 5, 7)
> 2*y
[1]  2  6 10 14
> crossprod(c(1,1,1), c(2,2,2)) # Actually dot
     [,1]
[1,]    6
>
> # Matrix Multiplication
```

```
> H <- matrix(c(1,2,2,1), nrow=2, ncol=2)
> H
     [,1] [,2]
[1,]    1    2
[2,]    2    1
>
> H%*%H
     [,1] [,2]
[1,]    5    4
[2,]    4    5
>
> # solutions of Equations
> A <- matrix(c(1,2,1,1), nrow=2, ncol=2)
> b <- c(1,2)
> solve(A,b)
[1] 1 0
> solve(A)
     [,1] [,2]
[1,]   -1    1
[2,]    2   -1
>
> # Set Ops
> x <- c(1,2,3,4)
> y <- c(3,4,5,6)
> union(x,y)
[1] 1 2 3 4 5 6
> intersect(x,y)
[1] 3 4
> 2%in%y
[1] FALSE
> choose (4,2)
[1] 6
```

CLASS PROBLEMS' OUTPUTS

```
> # Output should be 1, 2, 3 ... 15
> for(i in 1:15)
+ {
+  print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
```

```
[1] 13
[1] 14
[1] 15
>
> # Output should be 1, 3, 6 ... 120
> j=0
> for(i in 1:15)
+ {
+  j=j+i
+  print(j)
+ }
[1] 1
[1] 3
[1] 6
[1] 10
[1] 15
[1] 21
[1] 28
[1] 36
[1] 45
[1] 55
[1] 66
[1] 78
[1] 91
[1] 105
[1] 120
>
> # While Loop
> # Output should be 1, 2, 3 ... 15
> i=0
> while(i<15){
+  i=i+1
+  print(i)
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
>
> # Output should be 1, 3, 6 ... 120
```

```
> i=0
> j=0
> while(i<15){
+  i=i+1
+  j=j+i
+  print(j)
+ }
[1] 1
[1] 3
[1] 6
[1] 10
[1] 15
[1] 21
[1] 28
[1] 36
[1] 45
[1] 55
[1] 66
[1] 78
[1] 91
[1] 105
[1] 120

> # Repeat Loop
> # Output should be 1, 2, 3 ... 15
> i=1
> repeat {
+   print(i)
+   i=i+1
+   if(i>15){
+     break
+   }
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
[1] 13
[1] 14
[1] 15
> # Output should be 1, 3, 6 ... 120
> i=1
> j=1
```

```
> repeat{
+   print(j)
+   i=i+1
+   j=j+i
+   if(i>15){
+     break
+   }
+ }
[1] 1
[1] 3
[1] 6
[1] 10
[1] 15
[1] 21
[1] 28
[1] 36
[1] 45
[1] 55
[1] 66
[1] 78
[1] 91
[1] 105
[1] 120

> # Take two numbers as input and add, subtract, multiply, divide
> {
+ a = readline("Enter A: ");
+ b = readline("Enter B: ");
+ }
Enter A: 1
Enter B: 2

> a = as.numeric(a);

> b = as.numeric(b);

> # Standard Arithmetic Ops
> print(a+b);
[1] 3

> print(a-b);
[1] -1

> print(a*b);
[1] 2

> print(a/b);
[1] 0.5

> buildMatrix <- function() {
+  ro <- as.numeric(readline("Enter the number of rows: "))
```

```
+  cat("Enter the all the values row-wise: ")
+  val <- scan(stdin())
+  matrix(val, ro, byrow=TRUE)
+ }
> A <- buildMatrix()
Enter the number of rows: 2
Enter the all the values row-wise:
1: 1
2: 2
3: 3
4: 4
5:
Read 4 items
> B <- buildMatrix()
Enter the number of rows: 2
Enter the all the values row-wise:
1: 1
2: 0
3: 0
4: 1
5:
Read 4 items
> print(A+B)
     [,1] [,2]
[1,]    2    2
[2,]    3    5
> print(A-B)
     [,1] [,2]
[1,]    0    2
[2,]    3    3
> print(A%*%B)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```