**R LAB – 08**

**Task – 01 :**

**Aim :** Apply R math functions in a data or any dataset

**Program :**

# Doing Math With Simulation R

# Computing the probabilites of the given vector

exact\_one = function(p)

{

notp = 1 - p

tot = 0.0

for(i in 1:length(p))

{

tot = tot + p[i] + prod(notp[-i])

}

return(tot)

}

v1 = c(1,2,3,4,5)

a1 = exact\_one(v1)

a1

# Cumulative Sum and Product Of an Vector

cumsum(v1)

cumprod(v1)

# minima and maxima

mat1 <- matrix(c(1,5,6,2,3,2), nrow = 3)

mat1

min(mat1[,1], mat1[,2])

pmin(mat1[,1], mat1[,2])

pmin(mat1[1,], mat1[2,], mat1[3,])

# Sorting in R

unsort = c(13,2,5,2,3)

sort(unsort)

unsort

order(unsort) # it will get the indices of the sorted values in the original vector

# sorting the dataframe

v2 = c('def', 'ab', 'zzzz')

v3 = c(2,5,1)

y = data.frame(v1 = v2, v2 = v3)

y

r <- order(y$v2)

r

z <- y[r,]

z

rank(v3)

rank(v1)

# Linear ALgebra

crossprod(1:3,c(5,12,13)) # 1\*5 + 2\*12 + 3\*13 = 68

# Matrix Multiplication

a = matrix(1:4, ncol = 2, byrow = TRUE)

a

b = matrix(c(1,0,-1,1), nrow = 2)

b

mat\_mul = a %\*% b

mat\_mul

# solving the equations

s1 = matrix(c(1,1,-1,1), nrow = 2, ncol = 2)

s2 = c(2,4)

solve(s1,s2)

solve(s1)

# matrix Transpose

q1 = matrix(1:9, nrow = 3, byrow = TRUE)

q1

t(q1)

# determinent of matrix

det(q1)

# finding the eigen values

eigen(q1)

# Set operations in R

p <- c(1,2,5)

q <- c(5,1,8,9)

union(p,q)

intersect(p,q)

setdiff(p,q)

setdiff(q,p)

setequal(p,q)

setequal(p, c(1,2,5))

5 %in% p

10 %in% q

choose(5,3)

# Finding Symmetric Difference

symetric = function(a,b)

{

sx = setdiff(a,b)

sy = setdiff(b,a)

result = union(sx,sy)

return(result)

}

f = c(1,2,5)

g = c(5,1,8,9)

ans = symetric(f,g)

ans

**Output :**

exact\_one = function(p)

+ {

+ notp = 1 - p

+ tot = 0.0

+

+ for(i in 1:length(p))

+ {

+ tot = tot + p[i] + prod(notp[-i])

+ }

+

+ return(tot)

+ }

>

> v1 = c(1,2,3,4,5)

> a1 = exact\_one(v1)

> a1

[1] 39

> cumsum(v1)

[1] 1 3 6 10 15

> cumprod(v1)

[1] 1 2 6 24 120

> mat1 <- matrix(c(1,5,6,2,3,2), nrow = 3)

> mat1

[,1] [,2]

[1,] 1 2

[2,] 5 3

[3,] 6 2

> min(mat1[,1], mat1[,2])

[1] 1

> pmin(mat1[,1], mat1[,2])

[1] 1 3 2

> pmin(mat1[1,], mat1[2,], mat1[3,])

[1] 1 2

> unsort = c(13,2,5,2,3)

> sort(unsort)

[1] 2 2 3 5 13

> unsort

[1] 13 2 5 2 3

> order(unsort)

[1] 2 4 5 3 1

> v2 = c('def', 'ab', 'zzzz')

> v3 = c(2,5,1)

> y = data.frame(v1 = v2, v2 = v3)

> y

v1 v2

1 def 2

2 ab 5

3 zzzz 1

> r <- order(y$v2)

> r

[1] 3 1 2

>

> z <- y[r,]

> z

v1 v2

3 zzzz 1

1 def 2

2 ab 5

> rank(v3)

[1] 2 3 1

> rank(v1)

[1] 1 2 3 4 5

> crossprod(1:3,c(5,12,13))

[,1]

[1,] 68

> a = matrix(1:4, ncol = 2, byrow = TRUE)

> a

[,1] [,2]

[1,] 1 2

[2,] 3 4

> b = matrix(c(1,0,-1,1), nrow = 2)

> b

[,1] [,2]

[1,] 1 -1

[2,] 0 1

>

> mat\_mul = a %\*% b

> mat\_mul

[,1] [,2]

[1,] 1 1

[2,] 3 1

> s1 = matrix(c(1,1,-1,1), nrow = 2, ncol = 2)

> s2 = c(2,4)

> solve(s1,s2)

[1] 3 1

> solve(s1)

[,1] [,2]

[1,] 0.5 0.5

[2,] -0.5 0.5

> q1 = matrix(1:9, nrow = 3, byrow = TRUE)

> q1

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 4 5 6

[3,] 7 8 9

> t(q1)

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

> det(q1)

[1] 6.661338e-16

> eigen(q1)

eigen() decomposition

$values

[1] 1.611684e+01 -1.116844e+00 -1.303678e-15

$vectors

[,1] [,2] [,3]

[1,] -0.2319707 -0.78583024 0.4082483

[2,] -0.5253221 -0.08675134 -0.8164966

[3,] -0.8186735 0.61232756 0.4082483

> p <- c(1,2,5)

> q <- c(5,1,8,9)

> union(p,q)

[1] 1 2 5 8 9

> intersect(p,q)

[1] 1 5

>

> setdiff(p,q)

[1] 2

> setdiff(q,p)

[1] 8 9

>

> setequal(p,q)

[1] FALSE

> setequal(p, c(1,2,5))

[1] TRUE

>

> 5 %in% p

[1] TRUE

> 10 %in% q

[1] FALSE

>

> choose(5,3)

[1] 10

> symetric = function(a,b)

+ {

+ sx = setdiff(a,b)

+ sy = setdiff(b,a)

+ result = union(sx,sy)

+

+ return(result)

+ }

> f = c(1,2,5)

> g = c(5,1,8,9)

> ans = symetric(f,g)

> ans

[1] 2 8 9

**Result :** Successfully Executed the all lines in R