Name: Mithun G

USN: 19BTRCR006

LAB PROGRAM 10

1. Use R-function matrix to create the matrices called A and B:

a) Take the inverse of A and the transpose of A.
b) Multiply A with B.
c) Estimate the eigenvalues and eigenvectors of A.
d) For a matrix A, x is an eigenvector, and λ the eigenvalue of a matrix A, if A · x = λ · x. Test it!

In [1]:

```
A <- matrix(nrow=3, data=c(1,6,-2,2,4,1,3,1,-1))
B <- matrix(nrow = 3, data=c(1,2,-3,4,5,6,7,8,9))
print(A)
print(B)
```

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    6    4    1
[3,]   -2    1   -1
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]   -3    6    9
```

In [13]:

```
# taking the inverse
print(solve(A))
cat("\n")
# finding the transpose
print(t(A))
```

```
            [,1]       [,2]       [,3]
[1,] -0.11111111  0.1111111 -0.2222222
[2,]  0.08888889  0.1111111  0.3777778
[3,]  0.31111111 -0.1111111 -0.1777778

     [,1] [,2] [,3]
[1,]    1    6   -2
[2,]    2    4    1
[3,]    3    1   -1
```

In [15]:

```
# Multiply with A and B
A %*% B
```

A matrix: 3 × 3
of type dbl

-4	32	50
11	50	83
3	-9	-15

In [18]:

```
# Eigen values
ev <- eigen(A)
values <- ev$values
values
```

6.36669562473516+0i · -1.18334781236758+2.38069708993942i · -1.18334781236758-2.38069708993942i

In [19]:

```
# Eigen vectors
vectors <- ev$vectors
vectors
```

A matrix: 3 × 3 of type cpl

-0.36275602+0i	-0.0725936-0.5240033i	-0.0725936+0.5240033i
-0.93146469+0i	-0.2632991+0.4856291i	-0.2632991-0.4856291i
-0.02795726+0i	0.6441961+0.0000000i	0.6441961+0.0000000i

In [21]:

```
# A.x
A %*% vectors
```

A matrix: 3 × 3 of type cpl

-2.3095572+0i	1.333397+0.447255i	1.333397-0.447255i
-5.9303521+0i	-0.844561-1.201503i	-0.844561+1.201503i
-0.1779954+0i	-0.762308+1.533636i	-0.762308-1.533636i

In [23]:

```
# λ.X
values * vectors
```

A matrix: 3 × 3 of type cpl

-2.30955718+0.00000000i	-0.462181-3.336170i	-0.462181+3.336170i
1.10224670-2.21753527i	-0.844561-1.201503i	1.467710-0.052167i
0.03308317+0.06655777i	-0.762308-1.533636i	-0.762308-1.533636i

Answer: From the last 2 cells we can see that A·x = λ·x

2. Create a matrix, called P:
a) What is the value of the largest eigenvalue (the so-called dominant eigenvalue) and the corresponding eigenvector?
b) Create a new matrix, T, which equals P, except for the first row, where the elements are 0.
c) Now estimate N= (I-T)-1, where I is the identity matrix.

In [24]:

```
P <- matrix(nrow = 4, data = c(0,0.9775,0,0,0.0043,0.9111,0.0736,0,0.1132,0,0.9534,0.0452,0
P
```

A matrix: 4 × 4 of type dbl

0.0000	0.0043	0.1132	0.0000
0.9775	0.9111	0.0000	0.0000
0.0000	0.0736	0.9534	0.0000
0.0000	0.0000	0.0452	0.9804

In [25]:

```
# largest Eigen value
ev <- eigen(P)
values <- max(ev$values)
values
```

1.02544132553035

In [26]:

```
# largest Eigen vector
vectors <- max(ev$vectors)
vectors
```

1

In [32]:

```
# creating the new matrix T
T <- P
T[1,] <- 0
T
```

A matrix: 4 × 4 of type dbl

0.0000	0.0000	0.0000	0.0000
0.9775	0.9111	0.0000	0.0000
0.0000	0.0736	0.9534	0.0000
0.0000	0.0000	0.0452	0.9804

In [39]:

```
# Estimating (I-T)-1
N <- solve((diag(4)-T)-1)
N
```

A matrix: 4 × 4 of type dbl

0.7334085	-0.2687986	-0.27249196	-0.1959587
5.0654149	5.2694115	-6.06133692	-4.3589244
2.2794650	2.5542928	6.03849018	-11.0896043
-8.3448799	-7.8237044	0.02284673	15.4485287

3. Find the root of the equation ex = 4x2 in the interval [0, 1]. And draw the function curve.

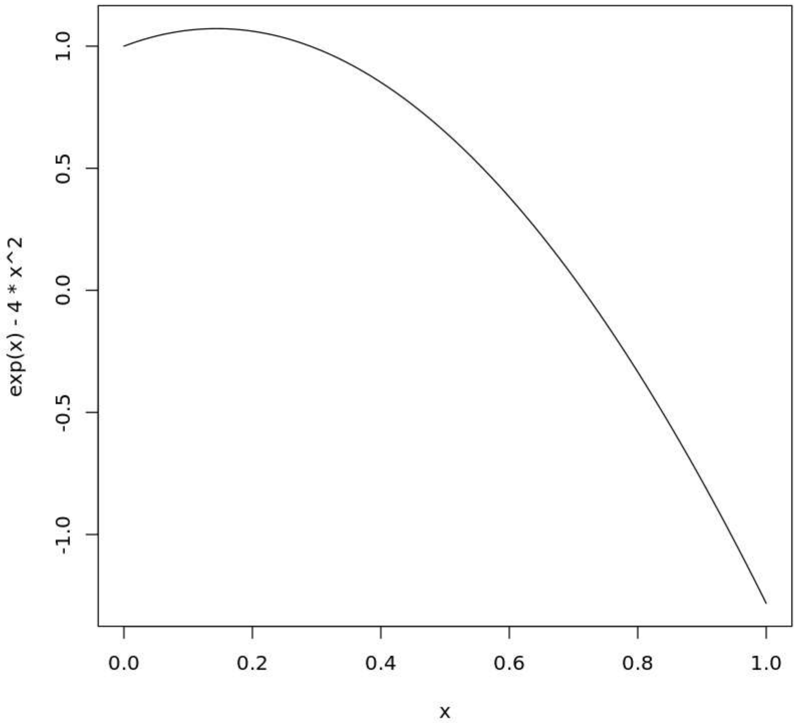
In [40]:

```
root <- uniroot(f=function(x) exp(x)-4*x^2,interval = c(0,1))
root
```

$root
0.714801396378604
$f.root
1.65946336081468e-05
$iter
5
$init.it
<NA>
$estim.prec
6.10351562503331e-05

In [41]:

```
curve(exp(x)-4*x^2,0,1)
```



4. Solve the equations 1000 = y ∗ (3 + x) ∗ (1 + y)4 for y and with x varying over the range from 1 to 100. Plot the root as a function of x.

In [42]:

```
res <- vector()
for(x in 1:100){
    res[x]<-uniroot(f=function(y) y*(3+x)*(1+y)^4-1000,c(-1000,1000))
}
```

```
“number of items to replace is not a multiple of replacement length”
Warning message in res[x] <- uniroot(f = function(y) y * (3 + x) * (1 + y)
^4 - 1000, :
“number of items to replace is not a multiple of replacement length”
Warning message in res[x] <- uniroot(f = function(y) y * (3 + x) * (1 + y)
^4 - 1000, :
“number of items to replace is not a multiple of replacement length”
Warning message in res[x] <- uniroot(f = function(y) y * (3 + x) * (1 + y)
^4 - 1000, :
“number of items to replace is not a multiple of replacement length”
Warning message in res[x] <- uniroot(f = function(y) y * (3 + x) * (1 + y)
^4 - 1000, :
“number of items to replace is not a multiple of replacement length”
Warning message in res[x] <- uniroot(f = function(y) y * (3 + x) * (1 + y)
^4 - 1000, :
“number of items to replace is not a multiple of replacement length”
Warning message in res[x] <- uniroot(f = function(y) y * (3 + x) * (1 + y)
^4 - 1000, :
“number of items to replace is not a multiple of replacement length”
```

In [43]:

```
plot(1:100,res,type = "l")
```