FE515 2022A Assignment 1

Lorenzo Ausiello

09/30/2023

Question 1:

1.1

Generate a vector x from 5 to 35 with increment 2, and calculate its length.

```
x<-seq(5,35,2)
x
```

```
##  [1]  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35
```

```
length(x)
```

```
## [1] 16
```

1.2

Use the vector x in 1.1 to generate a 4-by-4 matrix A which filled by rows.

```
A<-matrix(x,nrow=4,byrow=T)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    5    7    9   11
## [2,]   13   15   17   19
## [3,]   21   23   25   27
## [4,]   29   31   33   35
```

1.3

Calculate the eigenvalues of the matrix A in 1.2.

```
eigen(A)$values
```

```
## [1]  8.381780e+01 -3.817805e+00 -3.786132e-15  6.672852e-16
```

1.4

Change the 4 elements in first two rows and first two columns of the matrix A to 7. i.e. Let a11, a12, a21, a22 equal to 7.

```
A[1:2,1:2]=7
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7    7    9   11
## [2,]    7    7   17   19
## [3,]   21   23   25   27
## [4,]   29   31   33   35
```

1.5

Calculate the determinant of A in 1.4.

```
det(A)
```

```
## [1] 256
```

1.6

Calculate the inverse of A in 1.4.

```
A_inv<-solve(A)
A_inv
```

```
##        [,1]          [,2]    [,3]    [,4]
## [1,]  0.500 -6.278503e-16 -1.5000  1.0000
## [2,] -0.375 -1.250000e-01  1.3750 -0.8750
## [3,] -0.750  2.500000e-01 -0.4375  0.4375
## [4,]  0.625 -1.250000e-01  0.4375 -0.4375
```

1.7

Create a vector b by assigning the first row of A in 1.4 to b.

```
b<-A[1,]
b
```

```
## [1]  7  7  9 11
```

1.8

Find y by solving linear equation A y = b with the A in 1.4 and b in 1.7. (Hint. y can be found by y = A-1 b where A-1 is the inverse of A.)

```r
y<-solve(A,b)
y
```

```
## [1]  1.000 -0.750 -2.625  2.625
```

```r
#or
y2<-A_inv%*%b
y2
```

```
##        [,1]
## [1,]  1.000
## [2,] -0.750
## [3,] -2.625
## [4,]  2.625
```

1.9

For each element of y in 1.8 find the minimum between its value and pi/2. Store all results into a single vector. Print the value of resulting vector.

```r
vec<-c()
for (i in 1:length(y)){
  vec[i]<-min(y[i],pi/2)
  }
vec
```

```
## [1]  1.000000 -0.750000 -2.625000  1.570796
```

1.10

Read the documentation for function diag and use the function to generate the following 10-by-10 square matrix.

```r
?diag
```

```
## avvio in corso del server httpd per la guida ... fatto
```

```r
diag_values <- 1:10
square_matrix <- diag(diag_values)
square_matrix
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    0    0    0    0    0    0    0    0     0
##  [2,]    0    2    0    0    0    0    0    0    0     0
##  [3,]    0    0    3    0    0    0    0    0    0     0
##  [4,]    0    0    0    4    0    0    0    0    0     0
##  [5,]    0    0    0    0    5    0    0    0    0     0
##  [6,]    0    0    0    0    0    6    0    0    0     0
##  [7,]    0    0    0    0    0    0    7    0    0     0
##  [8,]    0    0    0    0    0    0    0    8    0     0
##  [9,]    0    0    0    0    0    0    0    0    9     0
## [10,]    0    0    0    0    0    0    0    0    0    10
```

Question 2

Consider a Fibonacci sequence Sn = Sn-1 + Sn-2 with the initial value S0 = 0 and the value at stage 1 as S1 = 1. Please determine the values of S3 and S50.

```
Sn=c(0,1)

for (i in 3:51) {
  Sn[i]=Sn[i-1]+Sn[i-2]
}

Sn
```

```
##  [1]           0           1           1           2           3           5
##  [7]           8          13          21          34          55          89
## [13]         144         233         377         610         987        1597
## [19]        2584        4181        6765       10946       17711       28657
## [25]       46368       75025      121393      196418      317811      514229
## [31]      832040     1346269     2178309     3524578     5702887     9227465
## [37]    14930352    24157817    39088169    63245986   102334155   165580141
## [43]   267914296   433494437   701408733  1134903170  1836311903  2971215073
## [49]  4807526976  7778742049 12586269025
```

```
S3<-Sn[4]
S50<-Sn[51]
c(S3,S50)
```

```
## [1]           2 12586269025
```

Question 3

Find all the integers between 1 and 100 which are divisible by both 3 and 5. Store the results into a vector. Print each element of the resulting vector.

```
vec2<-c()
for (i in 1:100){
  if(i%%3==0 & i%%5==0){
    vec2=c(vec2,i)
    print(i)
  }
}
```

```
## [1] 15
## [1] 30
## [1] 45
## [1] 60
## [1] 75
## [1] 90
```

Question 4

Create a function with input parameter n and returns a vector. The output vector contains all integers between 1 and n which are divisible by 3 and 5. Please test the function with two cases n = 100 and n = 200.

```
fun<-function(n){
  vec3<-c()
  for (i in 1:n){
    if(i%%3==0 && i%%5==0){
      vec3<-c(vec3,i)
    }
  }
  return(vec3)
}

fun(100)
```

```
## [1] 15 30 45 60 75 90
```

```
fun(200)
```

```
##  [1]  15  30  45  60  75  90 105 120 135 150 165 180 195
```

Question 5

Create a function with parameters a and b. In the function body, it tries to find the smallest positive number that is divisible by both a and b. Please test your function with following two cases (a = 3, b = 5) and (a = 6, b = 10).

```
fun2<-function(a, b) {
  c<-1
  while (c %% a != 0 | c %% b != 0) {
    c <- c + 1
  }
  return(c)
}

fun2(3,5)
```

```
## [1] 15
```

```
fun2(6,10)
```

```
## [1] 30
```

Question 6

Please find the attached JPM.csv and load this data into R. Make a subset of the loaded data frame. The subset contains open prices, closing prices and ad- justed closing prices (They are indicated by Open, Close and Adjusted). Please calculate the mean value of each column of the subset. (Hint. apply functions are better choice for this problem.)

```
setwd("C:/Users/loaus/OneDrive - stevens.edu/STEVENS/Intro to R/Assignment/Assignment_1")
data<-read.csv("FE515_hw1_JPM-2.csv")
head(data,5)
```

```
##   X JPM.Open JPM.High JPM.Low JPM.Close JPM.Volume JPM.Adjusted
## 1 1    48.00    48.37   47.59     48.07   14244700     32.52235
## 2 2    48.05    48.55   47.75     48.19    9471500     32.60353
## 3 3    48.17    48.25   47.63     47.79   10760500     32.33291
## 4 4    47.57    48.06   47.32     47.95    8239200     32.44115
## 5 5    47.90    48.11   47.36     47.75    9276700     32.30586
```

```
data_sub<-subset(data,select=c(JPM.Open,JPM.Close,JPM.Adjusted))
head(data_sub,5)
```

```
##   JPM.Open JPM.Close JPM.Adjusted
## 1    48.00     48.07     32.52235
## 2    48.05     48.19     32.60353
## 3    48.17     47.79     32.33291
## 4    47.57     47.95     32.44115
## 5    47.90     47.75     32.30586
```

```
sapply(data_sub,FUN=mean)
```

```
##     JPM.Open    JPM.Close JPM.Adjusted
##     72.50302     72.49895     62.64605
```