Homework1

2022-08-30

Question1

Consider the following system of equations:

```
3x + 2y + 2z + 4w = 28
2x+ y+ z =14
2x + 5z + 5w = 28
6x+2y+2z+ w=37
```

1. Create a matrix in R with the coefficients of the system, and a vector with the constants on the right-hand side of the equations. Call them mat1 and vec1, respectively.

```
mat1 <- matrix(c(3,2,2,4,2,1,1,0,2,0,5,5,6,2,2,1), nrow=4, ncol=4, byrow=TRUE);
vec1 <- matrix(c(28,14,28,37),nr=4,nc=1)
```

2. Find the inverse of mat1 and call it mat2.

```
mat2 <- solve(mat1)
```

3. Create a list named list1 having as components mat1, vec1, and mat2. Call these components item1, item2, and item3, respectively.

```
list1 <- list(item1= mat1, item2=vec1,item3=mat2)
```

4. Remove mat1, vec1, and mat2 from the working directory.

```
rm(mat1,vec1,mat2)
```

5. Solve the system of equations and call the solution vec2.

```
vec2 <- solve(list1[[1]], list1[[2]])
```

6. Verify the solution.

```
print(list1[[2]] - list1[[1]] %*% matrix(vec2) )
```

```
##               [,1]
## [1,]  0.000000e+00
## [2,]  0.000000e+00
## [3,] -3.552714e-15
## [4,]  0.000000e+00
```

You can see mat1 multiple vec2 equals vec1. So that the solution is correct.

7. Verify that if you multiply the inverse matrix mat2 by vec1 you get the solution.

```
print(vec2 - list1[[3]] %*%  list1[[2]] )
```

```
##               [,1]
## [1,] -2.664535e-15
## [2,]  3.996803e-15
## [3,]  1.776357e-15
## [4,]  0.000000e+00
```

You can see the inverse matrix mat2 by vec1 equals vec2. So that you get the solution.

8. Find the eigenvalues of mat1 and mat2 and verify that the eigenvalues of mat2 are the reciprocals of the eigenvalues of mat1.

```
mat1eigen <- eigen(list1[[1]])
mat2eigen <- eigen(list1[[3]])
m2v <- mat2eigen$values
m1v <- mat1eigen$values
print( rev(m1v) *m2v  )
```

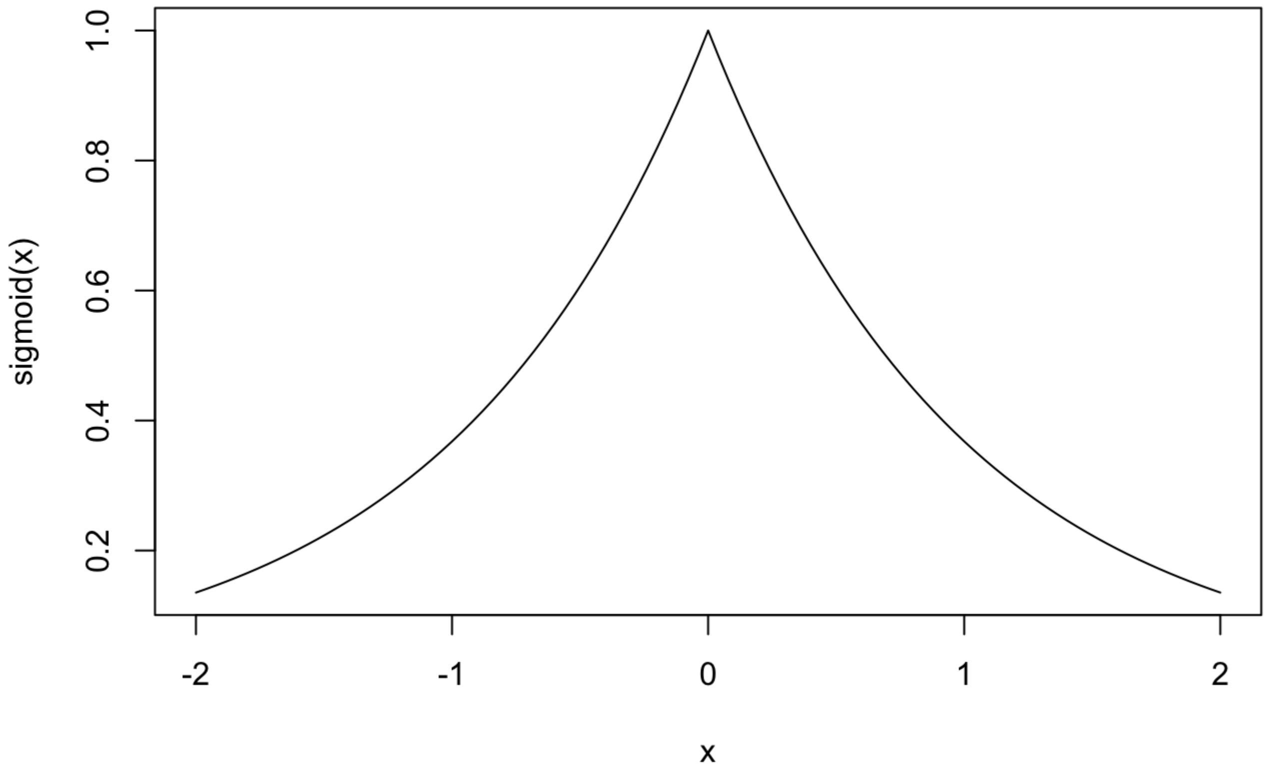
```
## [1] 1 1 1 1
```

Question2

Consider the function f(x) = e−|x|, for x ∈ R. We want to use the MonteCarlo method to estimate the value of the integral

1. Plot a graph of this function in the region where you want to calculate the integral.

```
sigmoid <- function(x) exp(-abs(x))
curve(sigmoid,-2,2)
```



2. Generate N = 1000 random numbers with uniform distribution in the rectangle [−2, 2] × [0, 1]. Count how many points fall below the curve f(x) = e−|x| and estimate the integral using the fraction of these points with respect to the total number of points and the area of the rectangle. Call the estimator I1000

the value of the integral = Area below the curve P(points fall below the curve) = Area below the curve/ rectangle area

```
x <- runif(1000,-2,2)
y <- runif(1000,0,1)
z <- exp(-abs(x))
asum <- sum(as.numeric(y<z))
recarea <- 4
I1000 <- recarea*asum/1000
print(I1000)
```

```
## [1] 1.716
```

3. Compute analytically the value of the integral and compare with the approximation you obtained in 3. Call I the value of the integral and calculate |I − I1000|

```
tmp <- integrate(sigmoid, lower = -2, upper = 2)
I <- tmp[[1]]
abs(I - I1000)
```

```
## [1] 0.01332943
```

```
results = list(abs(I - I1000))
summary(results)
```

```
##      Length Class  Mode
## [1,] 1      -none- numeric
```

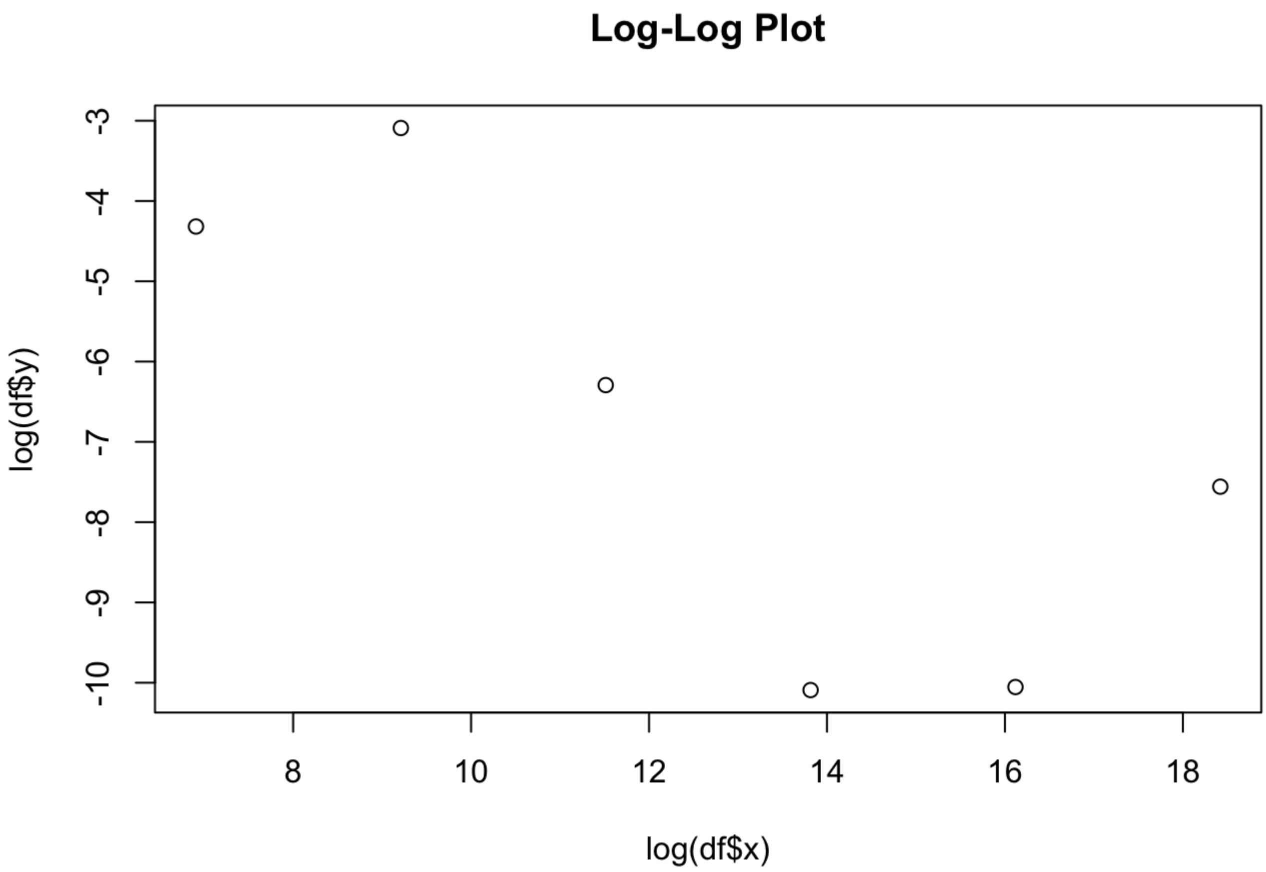
4. Repeat for N=10^k for k=4,5,…,8 and compute the deviation|I−I-N|from the exact result.

```
for (k in 4:8){
  x <- runif(10^k,-2,2)
  y <- runif(10^k,0,1)
  z <- exp(-abs(x))
  asum <- sum(as.numeric(y<z))
  recarea <- 4
  result <- 4*asum/10^k
  I <- tmp[[1]]
  print(abs(I - result))
  results <- append(results, abs(I - result))
}
```

```
## [1] 0.04547057
## [1] 0.001849434
## [1] 4.143353e-05
## [1] 4.303353e-05
## [1] 0.0005218065
```

5. Do a log-log plot of the deviation as a function of N. The points should follow approximately a straight line.

```
l =c(3:8)
myFun=function(x){10^x}
lis <- lapply(l,myFun)
a = as.vector(unlist(lis))
b = as.vector(unlist(results))
df <- data.frame(x=a, y=b)
plot(log(df$x), log(df$y), main='Log-Log Plot')
```


Log-Log Plot
log(df$y)
-3
-4
-5
-6
-7
-8
-9
-10
8
10
12
14
16
18
log(df$x)