INFO 534 Homework 1

Rachel Sagers

2023-01-27

Q1(a)

The dimension of the resulting matrix C from multiplying A times B will be 4 by 4. Matrix A is *mxn* (4,3), and matrix B is *nxo* (3,4), so the resulting matrix C will be *mxo* (4,4).

Q1(b)

The value of $C_{2,2}$ can be computed by multiplying the vector of row 2 of matrix A, by the vector of column 2 of matrix B and summing the products. The value of $C_{2,2}$ will thus be *5(3) + 1(1) + 2(1) = 18.*

Q1(c)

```r
# produce matrix A
matrixA <- matrix(c(4,5,-1,2,2,1,5,-3,2,2,1,3), nrow=4, ncol=3)

# produce matrix B
matrixB <- matrix(c(2,4,1,3,1,1,1,-3,-1,3,1,1), nrow=3, ncol=4)

# matrix multiplication
matrixC <- matrixA %*% matrixB

# resulting matrix C
print(matrixC)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   18   16   -4   16
## [2,]   16   18    0   18
## [3,]   19    3  -17    3
## [4,]   -5    6    8    6
```
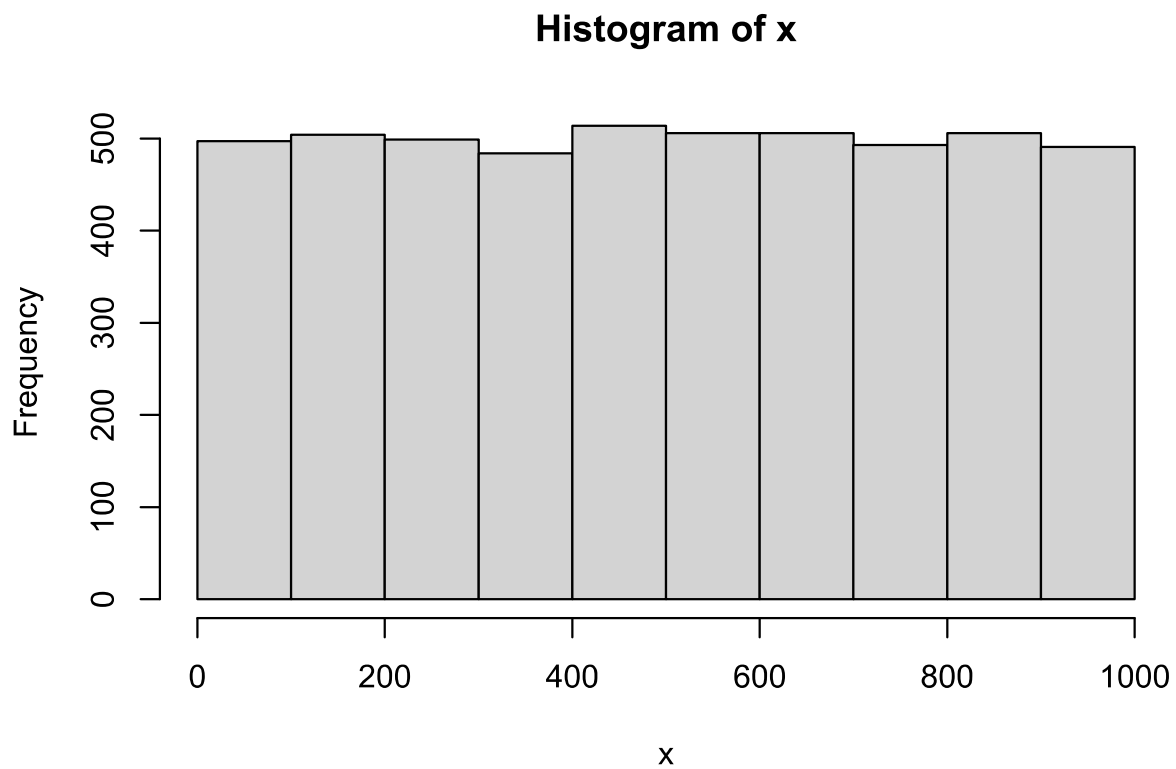
Q2

```r
# while loop
v <- 10
while(15 - v > 0.1)
{
  v <- v + runif(n=1, min=0, max=1)
  print(v)
}
```

```
## [1] 10.83823
## [1] 11.2475
```

```
## [1] 11.88288
## [1] 12.84692
## [1] 13.27478
## [1] 14.22318
## [1] 14.99784
```

Q3(a)

```r
# generate 5000 random values of x and create histogram
x <- runif(n=5000, min=0, max=1000)
hist(x)
```
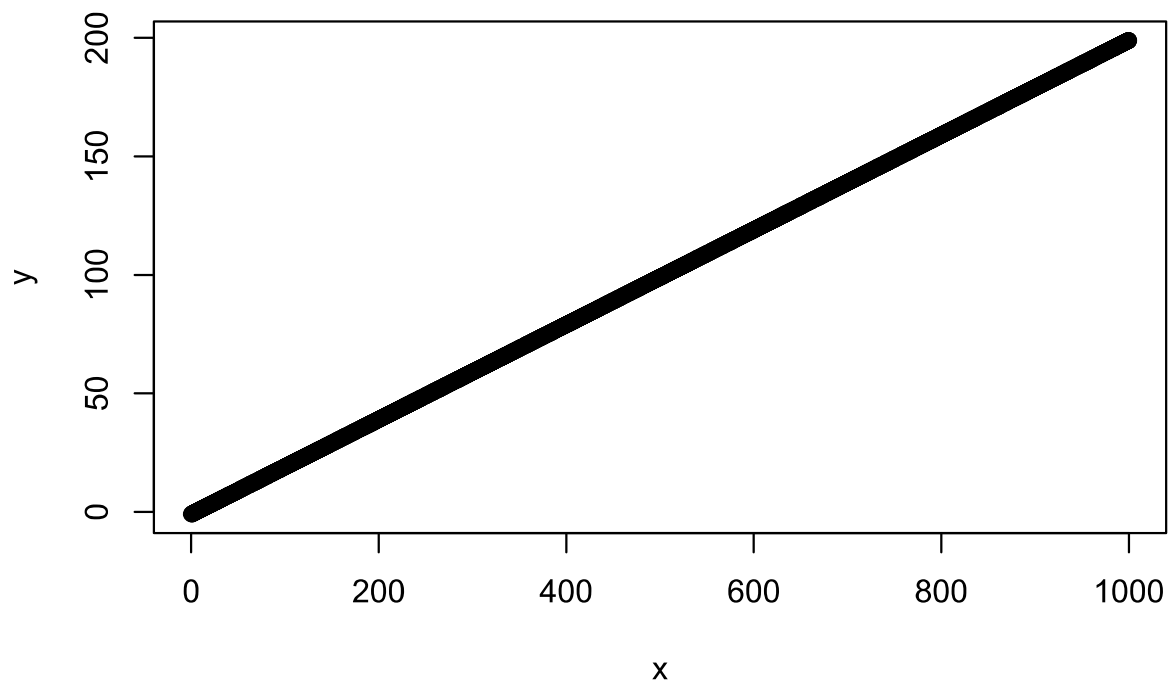
Histogram of x

Q3(b)

```r
# linear model function
output = function(a,x){
y = -1 + a*x
return(y)
}

# generate y values from x values using function
y <- output(x=x, a=0.2)

# plot of x vs y
plot(x,y)
```

2

Q4(a)

```r
# the range of the model parameter value to be considered
a.values = seq(-2,2,0.1)

# log likelihood function
logL = function(a, x, y){
  sigma=1
  500*log(1/(sqrt(2*pi)*sigma)) - 1/(sigma^2)*(1/2)*sum((a*x - y)^2)
}

# create an empty vector to store output of for loop
logoutput <- numeric()

# for loop to input all a values in function logL
for(a in a.values) {
  output <- logL(a=a, x=x, y=y)
  logoutput <- c(logoutput, output)
}

# plot of a values vs logL function output
plot(a.values, logoutput)
```
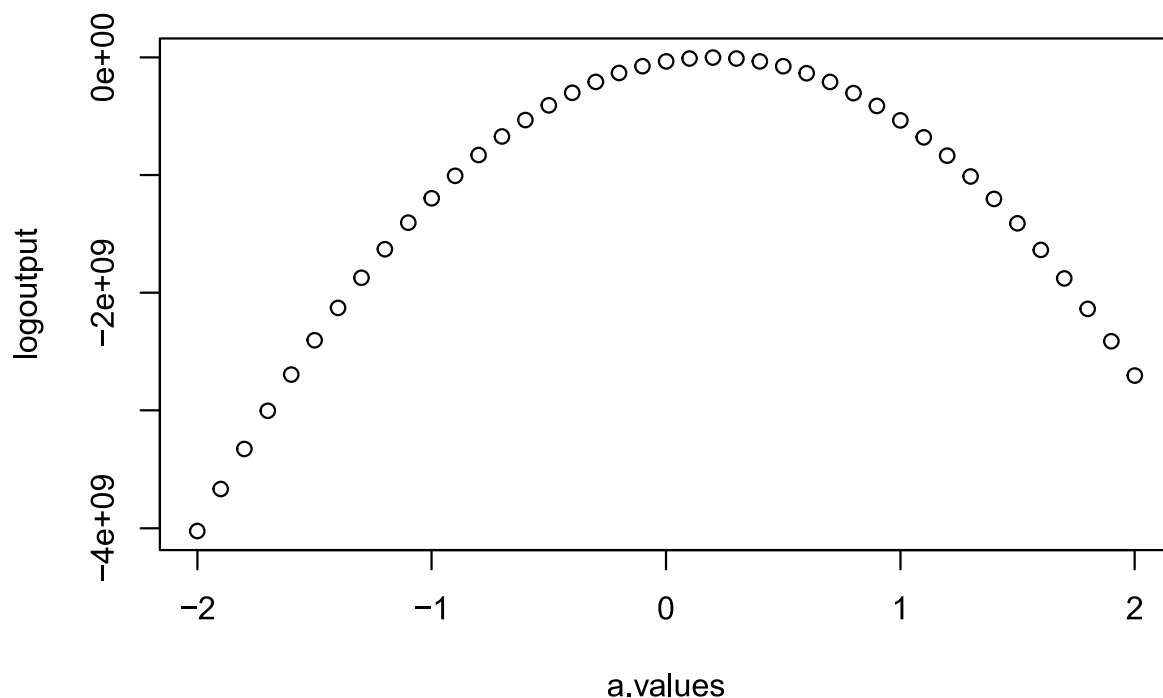
By visually inspecting the resulting graph, I would estimate that the value of a that corresponds to the highest value of the log-likelihood is about **0.4**.

Q4(b)

```r
optim(par=0, logL, x=x, y=y,method = "L-BFGS-B",control = list(fnscale=-1))
```

```
## $par
## [1] 0.1984995
##
## $value
## [1] -1084.462
##
## $counts
## function gradient
##        7        7
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

If the function argument *fnscale* is set to -1, it will lead to a maximization of the defined log-likelihood.

Q4(c)

4

```
mleresult <- optim(par=0.5, logL, x=x, y=y,method = "L-BFGS-B",control = list(fnscale=-1))
print(mleresult$value)
```

```
## [1] -1084.462
```

Yes, the value of the MLE *-1080.794* with the input parameter *0.5* is consistent with the answer in question 4(a).