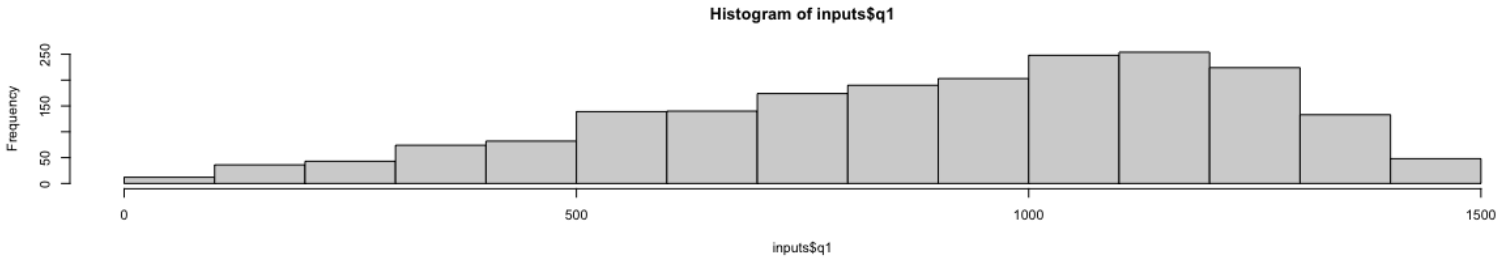
Valeria Aybar

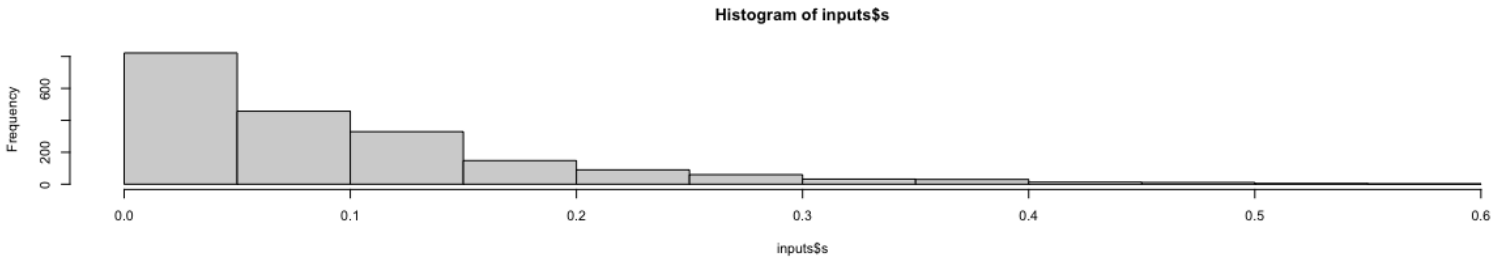
CAP4830 HW3

1) Create a data frame named **inputs** with the following column names (q1, q2, p, s) and each column has 2000 rows with the following data:

- q1 -> 2000 random variables that have a triangle distribution with A = 0, B = 1500 , C = 1200
- q2 -> 2000 random variables that have a triangle distribution with A = 0, B = 3500 , C = 1000
- p -> 2000 random variables that have a triangle distribution with A = 10, B = 17.50 , C = 12.50
- s -> 2000 random variables that have an exponential distribution with l =10

2) Plot the histogram in a single window of each column of the **inputs** data frame. Hint use par(mfrow=c(4, 1))

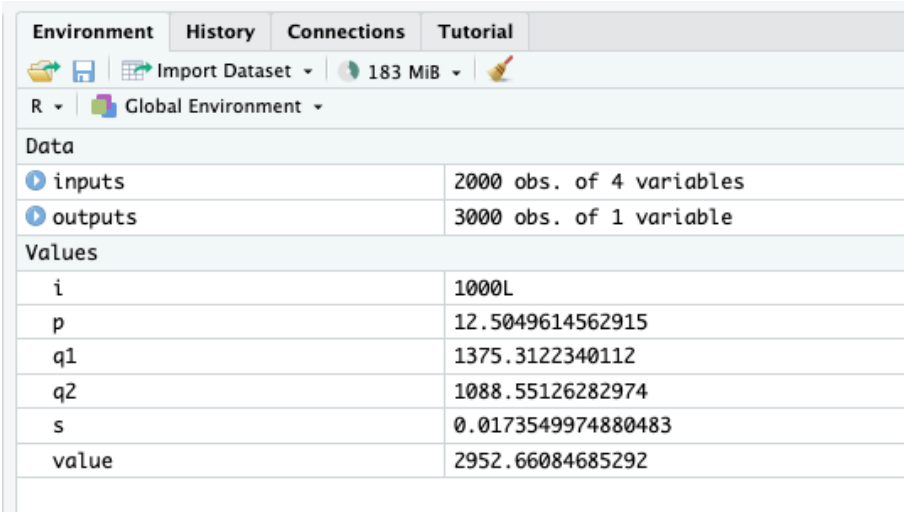




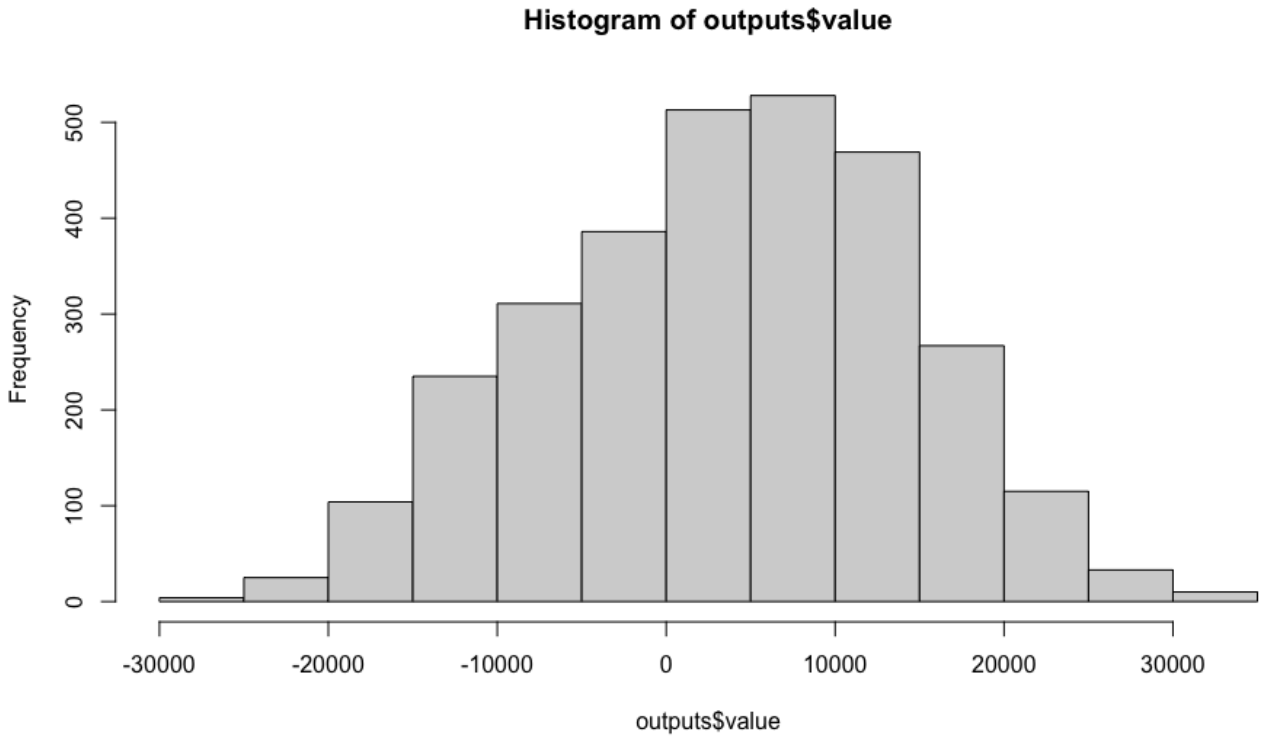
3) Create a data frame named **outputs** with a column name **value** that stores the output of the following model:

$$f(q1, q2, p, s) = (2700 – q1\text{-}q2) *p – (s*p)$$

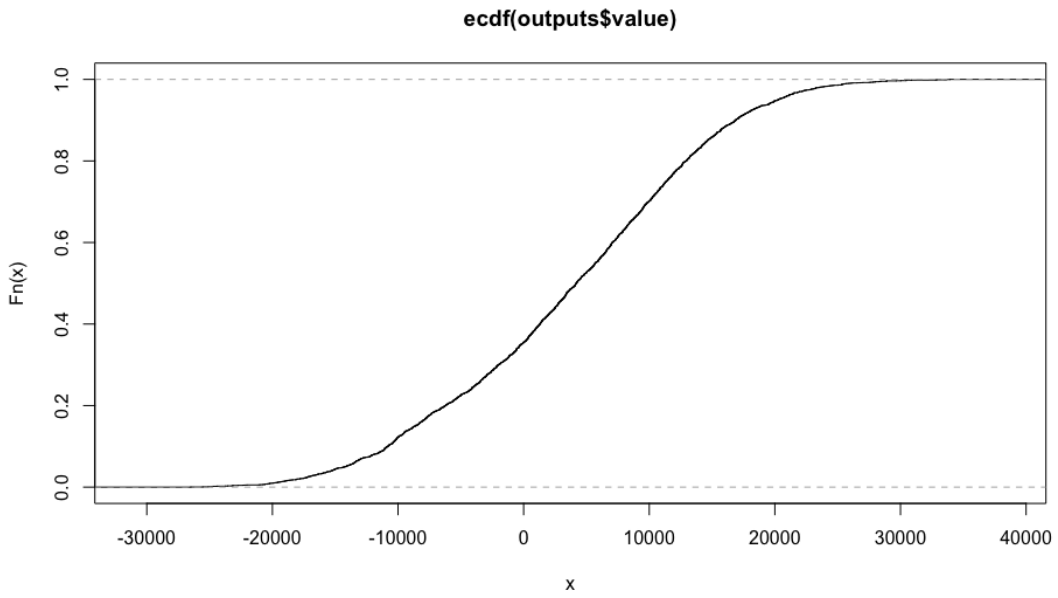
4) Run a Monte Carlo Simulation 1000 times of the model shown in #3, and store the results in the **value** column of the data frame **outputs**



5) Plot the histogram of outputs$value



6) Create the empirical CDF of outputs$value and plot this CDF



7) Calculate the P0, P10, P20, P30, … P90, P100 and output these values on the R-Console

```
60  #7 calculate the P0,P10... and output values on R-Console
61  quantile(outputs$value, probs = c(seq(0,1,by = 0.1)))
62
63  #8 find P20 to P80 interval values
64
65  px <- data.frame (values = quantile(outputs$value, probs = c(seq(0,1,by = 0.1))))
```

61:54 (Untitled) R Script

Console Terminal Background Jobs

R 4.2.2 · ~/

```
+
+   # store the results with outputs
+   q1 <- sample(inputs$q1 , 1)
+   q2 <- sample(inputs$q2 , 1)
+   p <- sample(inputs$p , 1)
+   s <- sample(inputs$s , 1)
+
+   value <- ((2700 - q1 - q2) * p - (s*p))
+
+   # store the results within outputs
+   outputs <- rbind(outputs, data.frame(value))
+ }
> hist(outputs$value)
> hist(outputs$value)
> empiricalCDF <- ecdf(outputs$value)
> plot(empiricalCDF)
> quantile(outputs$value, probs = c(seq(0,1,by = 0.1)))
        0%        10%        20%        30%        40%        50%        60%        70%        80%        90%       100%
-26495.888 -10836.816  -6206.979  -1997.033   1304.400   4190.421   7060.896   9940.221  12922.506  16884.749  33940.706
>
```

8) Find the P20 to P80 interval values from the empirical CDF.

```
63  #8 find P20 to P80 interval values
64
65  px <- data.frame (values = quantile(outputs$value, probs = c(seq(0,1,by = 0.1))))
66
67  sprintf("Interval of Interest: [ %.2f , %.2f ]",  px$values[3], px$values[9])
68
69  #9 create matrix named storage that stores 100 samples
70
71  storage <- matrix(ncol = 100, nrow = 250)
72
```

68:1 (Untitled) R Script

Console Terminal Background Jobs

R 4.2.2 · ~/

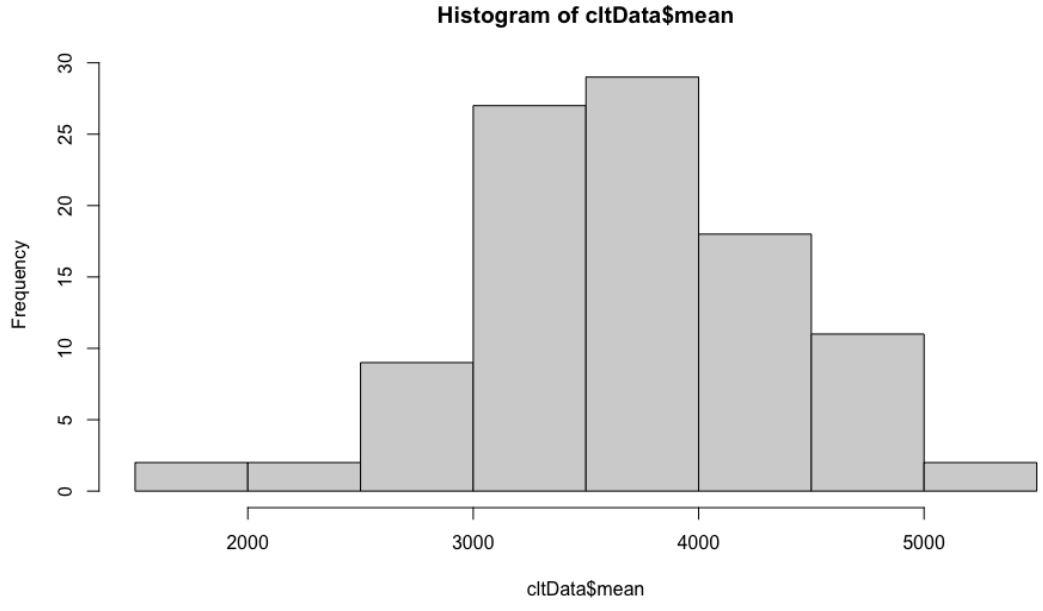
```
+   p <- sample(inputs$p , 1)
+   s <- sample(inputs$s , 1)
+
+   value <- ((2700 - q1 - q2) * p - (s*p))
+
+   # store the results within outputs
+   outputs <- rbind(outputs, data.frame(value))
+ }
> hist(outputs$value)
> hist(outputs$value)
> empiricalCDF <- ecdf(outputs$value)
> plot(empiricalCDF)
> quantile(outputs$value, probs = c(seq(0,1,by = 0.1)))
       0%        10%        20%        30%        40%        50%        60%        70%        80%        90%       100%
-26495.888 -10836.816  -6206.979  -1997.033   1304.400   4190.421   7060.896   9940.221  12922.506  16884.749  33940.706
> px <- data.frame (values = quantile(outputs$value, probs = c(seq(0,1,by = 0.1))))
>
> sprintf("Interval of Interest: [ %.2f , %.2f ]",  px$values[3], px$values[9])
[1] "Interval of Interest: [ -6206.98 , 12922.51 ]"
>
```

9) Create a matrix named **storage** that stores 100 samples of the Monte Carlo simulation of the model in #3. Each Monte Carlo simulation will have 250 realizations.

10) Convert the **storage** matrix into a data frame named **storage.**

11) Create a data frame named **cltData** that stores all the means of each column of the **storage** data frame.

12) Plot the histograms of the data within **cltData**



13) Check if the data within **cltData** is normally distributed. Hint use a normal test for this.

```
102  #13 Use Shapiro test to check if cltdata is normally distributed
103
104  shapiro.test(cltData$mean)
105
106  #14 state if the data in cltData is normally distributed
```

104:1 (Untitled)

Console | Terminal | Background Jobs

R 4.2.2 · ~/

```
+     s <- sample(inputs$s , 1)
+
+     value <- ((2700 - q1 - q2) * p - (s*p))
+
+     outputData <- rbind(outputData, data.frame(value))
+   }
+   storage[ , i] <- outputData$value
+ }
> storage <- data.frame(storage)
> cltData <- data.frame(mean = colMeans(storage))
> par(mfrow=c(1,1))
> hist(cltData$mean)
> shapiro.test(cltData$mean)

        Shapiro-Wilk normality test

data:  cltData$mean
W = 0.98514, p-value = 0.3252
```

14) State in your code using a comment if the data in cltData is normal distributed or not and why.

```
106  #14 state if the data in cltData is normally distributed
107
108  sprintf("Considering that the p-value is %.4f it is less than 0.5, which means its significantly different from normal distri
109          shapiro.test(cltData$mean)$p)
110
111  #15 calculate the 80% confidence interval and output the lower and upper bounds
112
```

110:1 (Untitled) R Script

Console | Terminal | Background Jobs

R 4.2.2 · ~/

```
> shapiro.test(cltData$mean)

        Shapiro-Wilk normality test

data:  cltData$mean
W = 0.98514, p-value = 0.3252

> sprintf("Considering that the p-value is %.4f it is less than 0.5, which means its significantly different from normal distributio
n.",
+         shapiro.test(cltData$mean)$p)
[1] "Considering that the p-value is 0.3252 it is less than 0.5, which means its significantly different from normal distribution."
>
```

15) Calculate the 80% confidence internal and output the lower and upper bounds of this interval on the R-console