MSAN 593 - Homework 2

Andre Guimaraes Duarte

July 24, 2016

Question 1

1.1

1.1.1

```
myRunifVec <- runif(10000000, 4, 6)
hist(myRunifVec,
     main = paste("Histogram of ", length(myRunifVec), "\n random variables ~ U(4, 6)",
                  sep = ""), xlab = "x", freq = F)
```

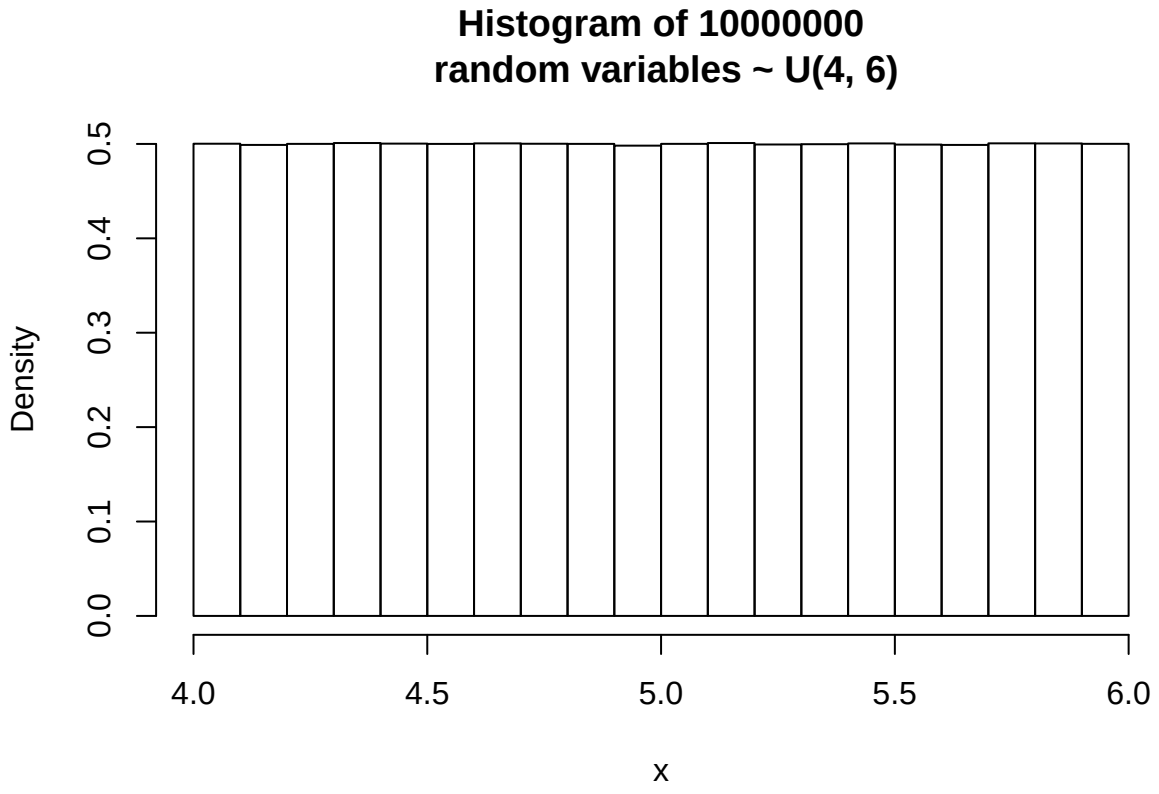

Figure 1: Histogram of 10,000,000 random variables ~ U(4, 6)

The histogram in figure 1 is the histogram of a $\sim U(4, 6)$ distribution with 10000000 random variables.

1.1.2

```r
samples <- 100000
myRunifSample <- sample(myRunifVec, 100000)
hist(myRunifSample,
     main = paste("Histogram of ", length(myRunifSample),
                  "\n random variables sampled from a ~ U(4, 6)",
                  sep = ""), xlab = "x", freq = F)
```

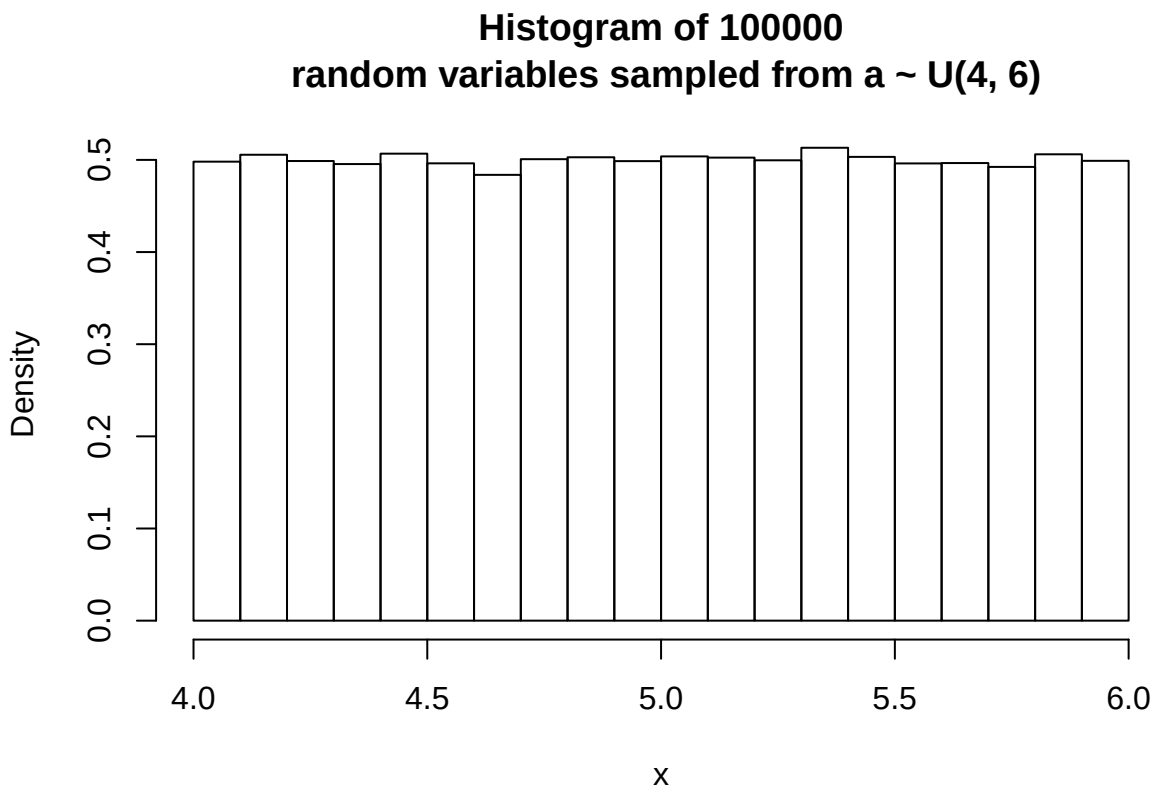

Figure 2: Histogram of a sample of 100,000 random elements from a ~U(4, 6) distribution

The histogram in figure 2 is very similar to the population distribution 1. Sampling elements from a uniform distribution maintains its original distribution.

1.1.3

Disclaimer: I first did this assignment in its entirety using `for` loops. My computer handled up to 1,000 samples fairly OK, but once the number of samples reached a few tens of thousands, it was unbearably slow to take the samples and means. I figured that `for` loops were not the way to go for this exercise, and did some research of some vectorized alternatives. I found that using the `apply` method was conveninent in this case, and indeed the results are much, **much** faster.

First, I define a data frame called `myRunifDataFrame` that has 30 columns. Each column will have `100,000` rows of single random samples from the vector `myRunifVec`. To do 30 repetitions of sampling 100,000 random variables, I use the `replicate` method. This does not simply copy the results from one column to the other, but indeed each column will have a separate sample of 100,000 random variables from the original distribution. `replicate` replicates the function, in this case `sample`.

Then, I define a second data frame, called `myRunifMeans` that contains the means of two, five, ten, and thirty elements from the original distribution. By using the `apply` method, I can define which columns (how many)

of `myRunifDataFrame` to use in order to compute the mean. This takes only a couple of seconds to run, opposed to many minutes by using the previous method (`for` loops) and is much more efficient.

```
myRunifDataFrame <- data.frame(replicate(30, sample(myRunifVec, 100000)))
myRunifMeans <- data.frame("Means2"= apply(myRunifDataFrame[,1:2], 1, mean),
                           "Means5"= apply(myRunifDataFrame[,1:5], 1, mean),
                           "Means10"= apply(myRunifDataFrame[,1:10], 1, mean),
                           "Means30"= apply(myRunifDataFrame[,1:30], 1, mean))
```

```
hist(myRunifMeans$Means2, main="Histogram of the average of two elements from a ~Unif(4, 6)",
     xlab="x", freq = F)
```

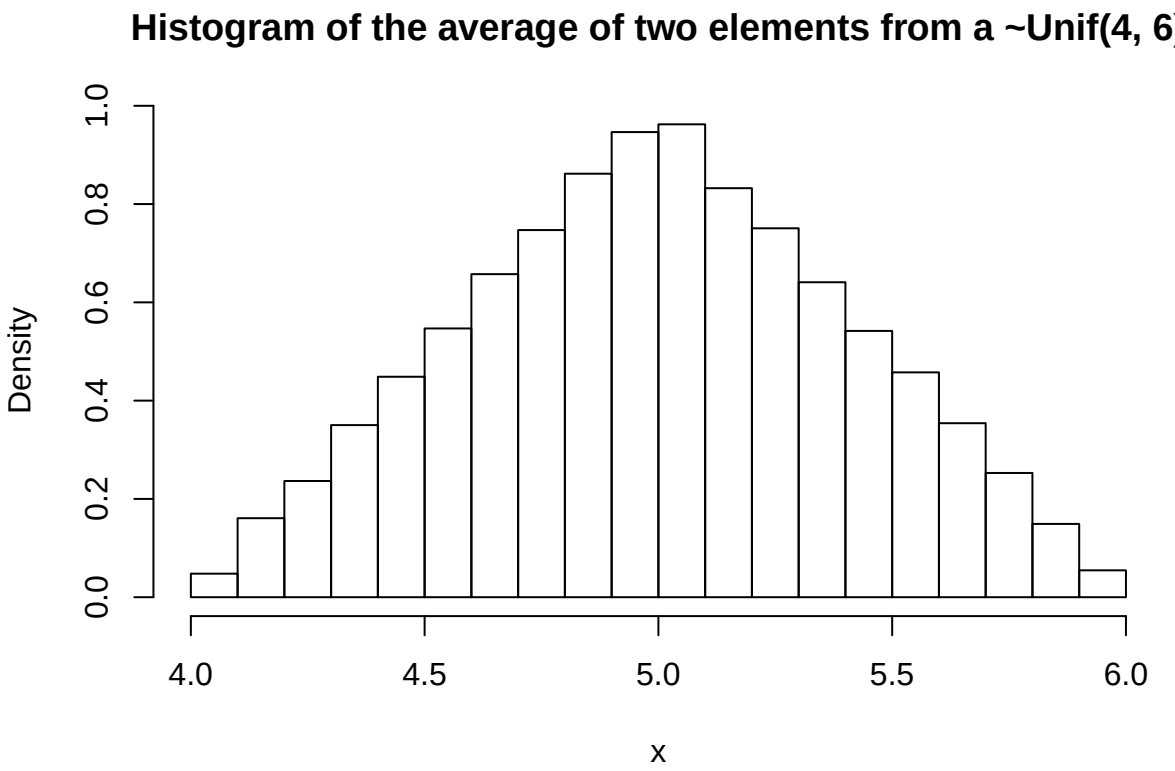

Figure 3: Histogram of 100,000 random means of two elements from a ~U(4, 6) distribution

The histogram in figure 3 is not similar to the population distribution shown in figure 1. In fact, the distribution seems very triangular.

1.1.4

```
hist(myRunifMeans$Means5, main="Histogram of the average of five elements from a ~Unif(4, 6)",
     xlab="x", freq = F)
```

The histogram in figure 4 is different from the population distribution in figure 1. There are more observations around 5, and less toward the tails. It is different than the previous histogram (figure 3): the distribution is less triangular.

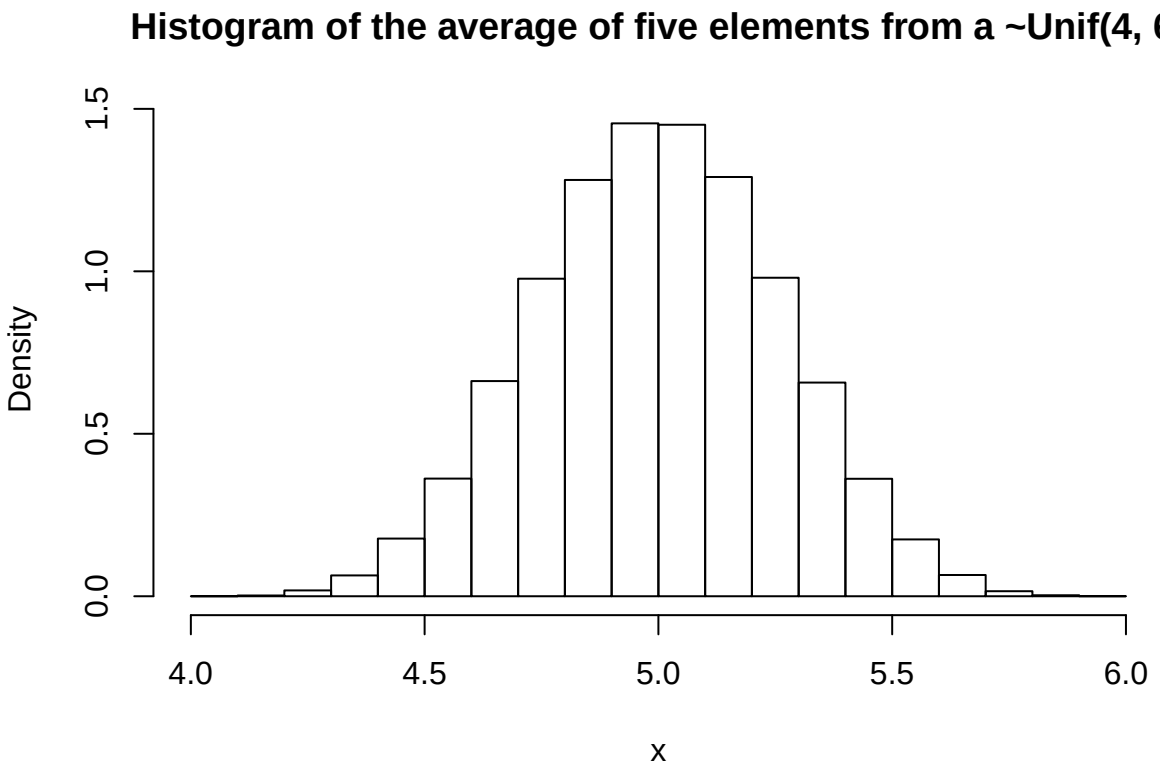

Figure 4: Histogram of 100,000 random means of five elements from a ~U(4, 6) distribution

1.1.5

```
hist(myRunifMeans$Means10, main="Histogram of the average of ten elements from a ~Unif(4, 6)",
     xlab="x", freq = F)
```

The histogram in figure 5 is different from the population distribution in figure 1. This one is bell-shaped and seems to be symmetrical around the value 5.

1.1.6

```
hist(myRunifMeans$Means30, main="Histogram of the average of thirty elements from a ~Unif(4, 6)",
     xlab="x", freq = F)
```

The histogram in figure 6 is different from the population distribution in figure 1. It looks a lot like a normal distribution centered around 5.

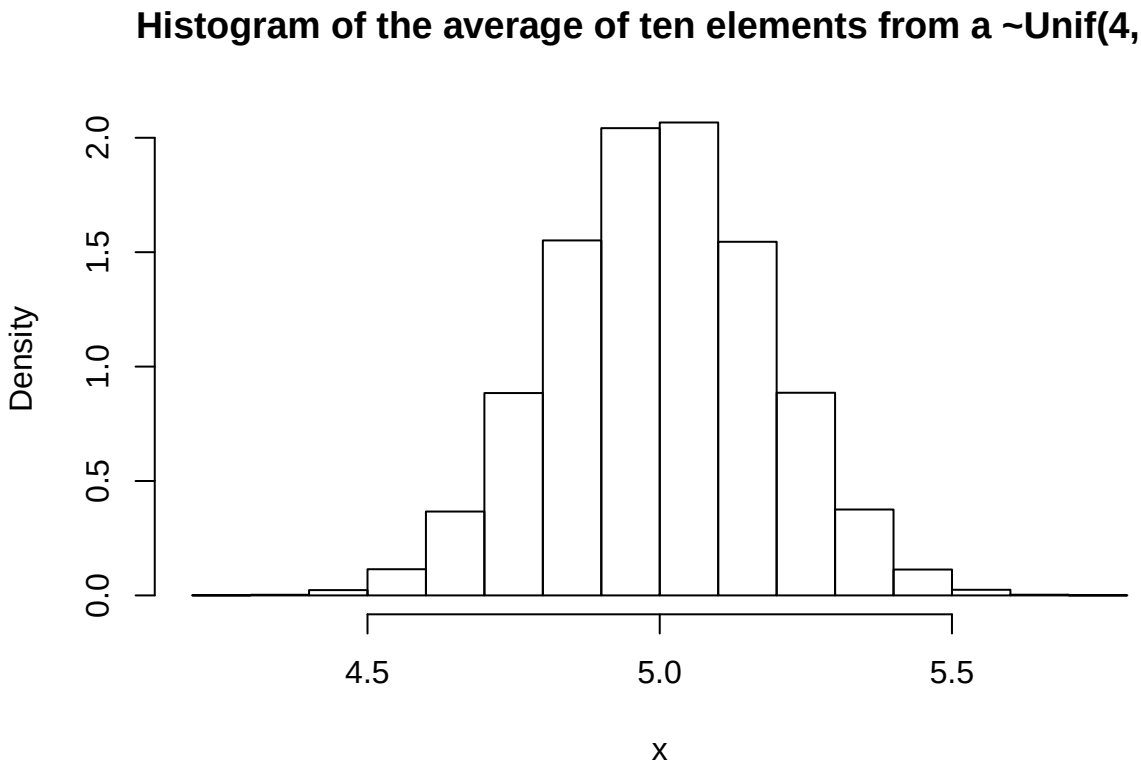

Figure 5: Histogram of 100,000 random means of ten elements from a ~U(4, 6) distribution

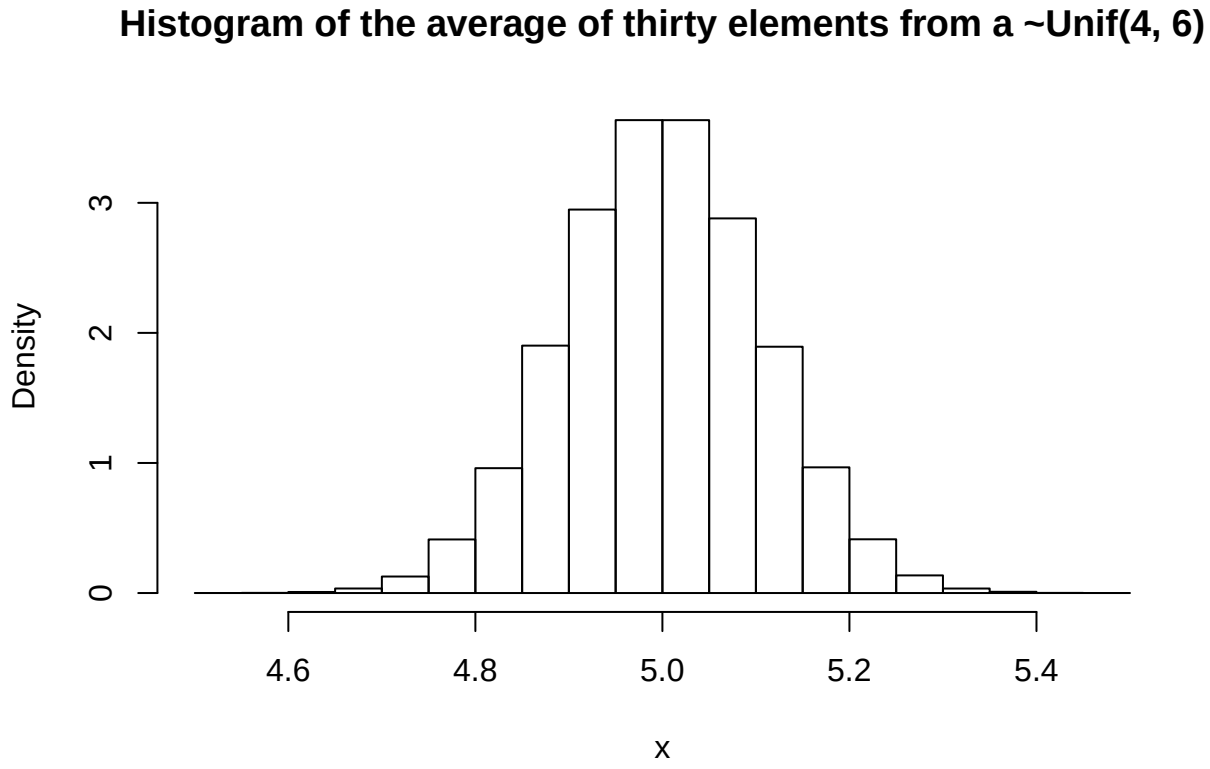

Figure 6: Histogram of 100,000 random means of thirty elements from a ~U(4, 6) distribution

1.2

1.2.1

```
myRexpVec <- rexp(10000000, 0.5)
hist(myRexpVec,
     main = paste("Histogram of ", length(myRexpVec), "\n random variables ~ Exp(0.5)",
                  sep = ""), xlab = "x", freq = F)
```

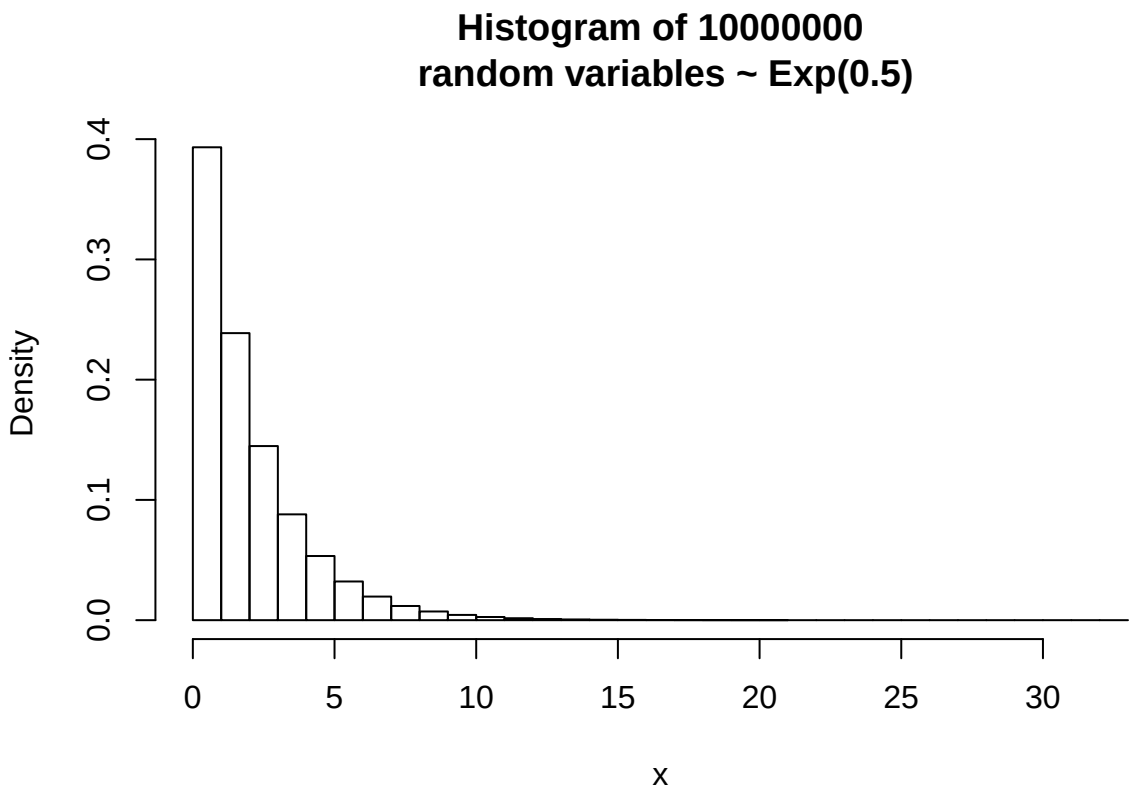

Figure 7: Histogram of 10,000,000 random variables ~Exp(0.5)

Figure 7 shows the histogram of a $\sim Exp(0.5)$ distribution with 10000000 random variables.

1.2.2

```
myRexpSample <- sample(myRexpVec, 100000)
hist(myRexpSample,
     main = paste("Histogram of ", length(myRexpSample),
                  "\n random variables sampled from a ~ Exp(0.5)",
                  sep = ""), xlab = "x", freq = F)
```

The histogram in figure 8 is very similar to the population distribution 7. Sampling elements from a negative exponential distribution maintains its original distribution.

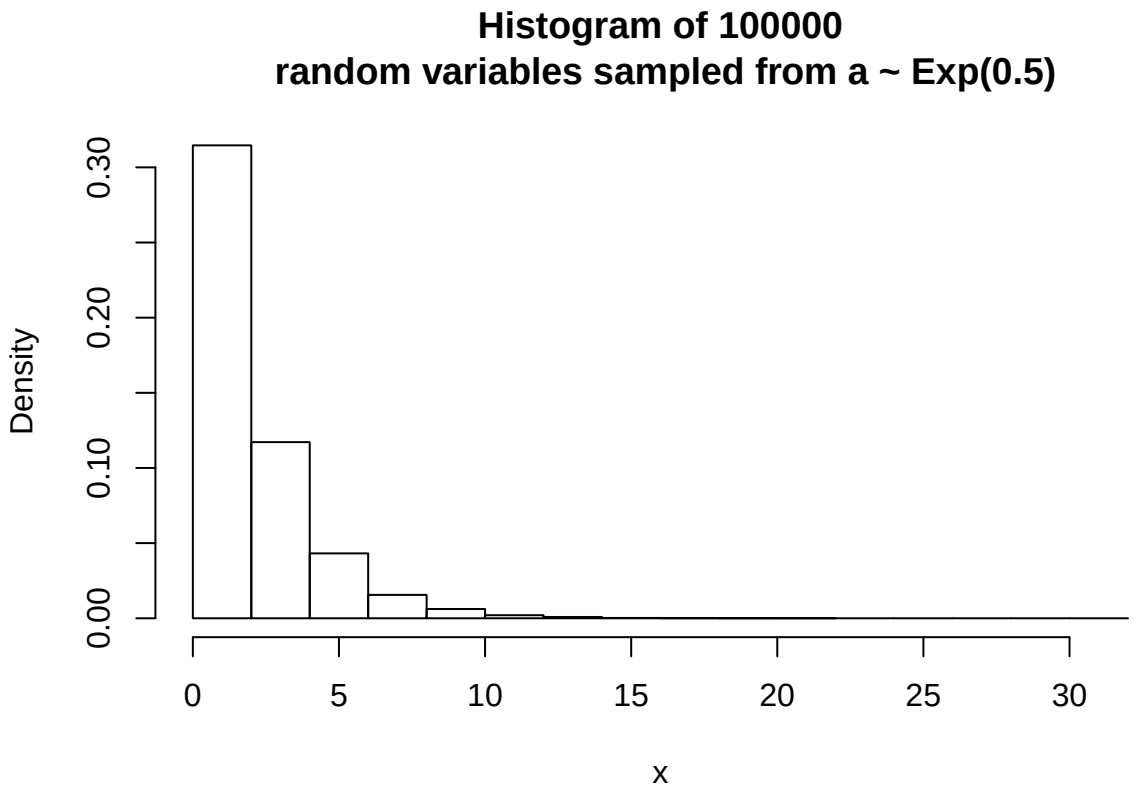

Figure 8: Histogram of 100,000 random elements from a ~Exp(0.5) distribution

1.2.3

```
myRexpDataFrame <- data.frame(replicate(30, sample(myRexpVec, 100000)))
myRexpMeans <- data.frame("Means2"= apply(myRexpDataFrame[,1:2], 1, mean),
                          "Means5"= apply(myRexpDataFrame[,1:5], 1, mean),
                          "Means10"= apply(myRexpDataFrame[,1:10], 1, mean),
                          "Means30"= apply(myRexpDataFrame[,1:30], 1, mean))
```

```
hist(myRexpMeans$Means2, main="Histogram of the average of two elements from a ~Exp(0.5)",
     xlab="x", freq = F)
```

Figure 9 shows the histogram is slightly more skewed to the right than the population distribution in figure 7. In fact, this is a Gamma distribution with parameter $\alpha = 2 * 0.5 = 1$.

1.2.4

```
hist(myRexpMeans$Means5, main="Histogram of the average of five elements from a ~Exp(0.5)",
     xlab="x", freq = F)
```

The histogram when we take the mean of five elements from the original population, shown in figure 10, is very different from the histogram of the population distribution in figure 7. It looks like the values are slowly distributing themselves around 2, but with a longer tail to the right.

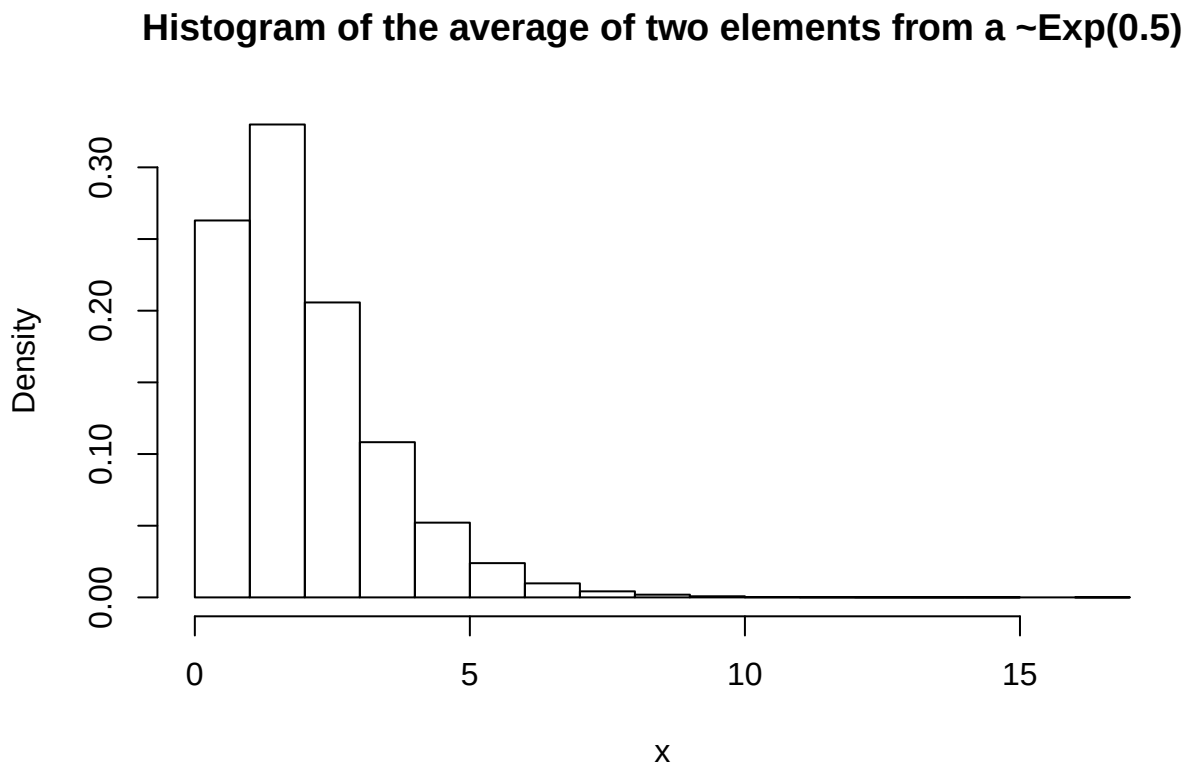

Figure 9: Histogram of 100,000 random means of two elements from a ~Exp(0.5) distribution

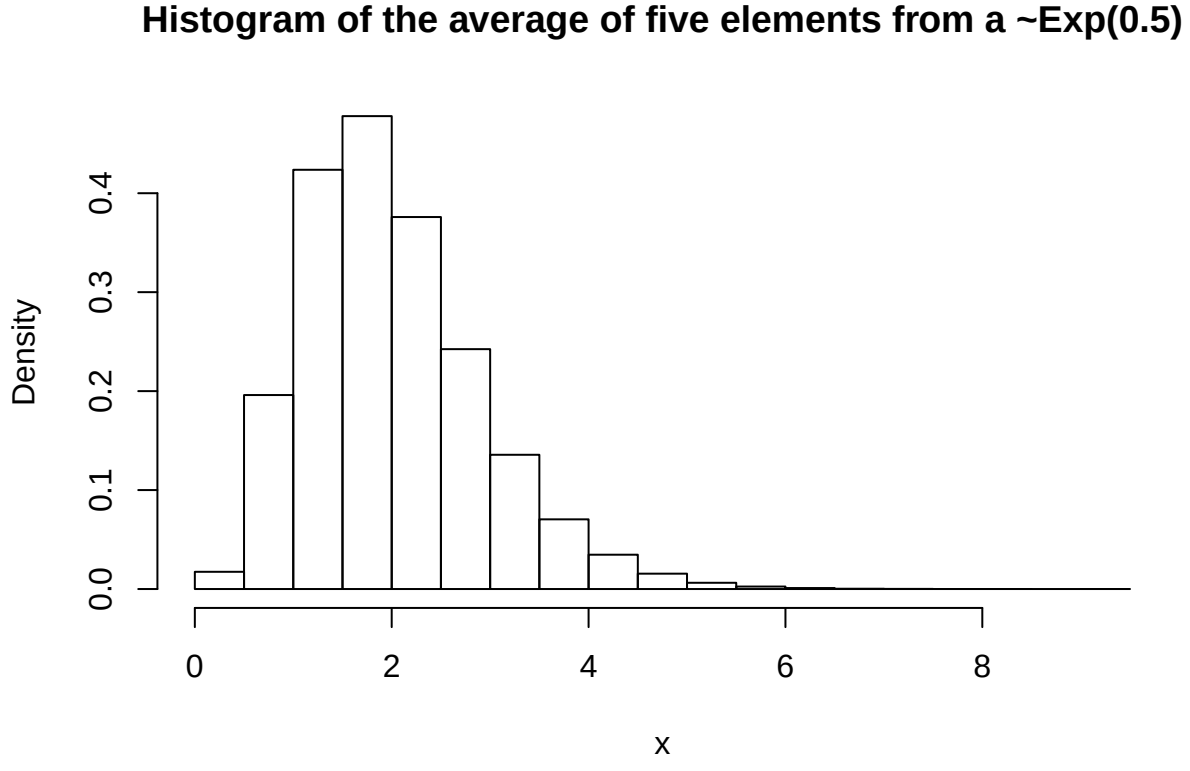

Figure 10: Histogram of 100,000 random means of five elements from a ~Exp(0.5) distribution

1.2.5

```
hist(myRexpMeans$Means10, main="Histogram of the average of ten elements from a ~Exp(0.5)",
     xlab="x", freq = F)
```

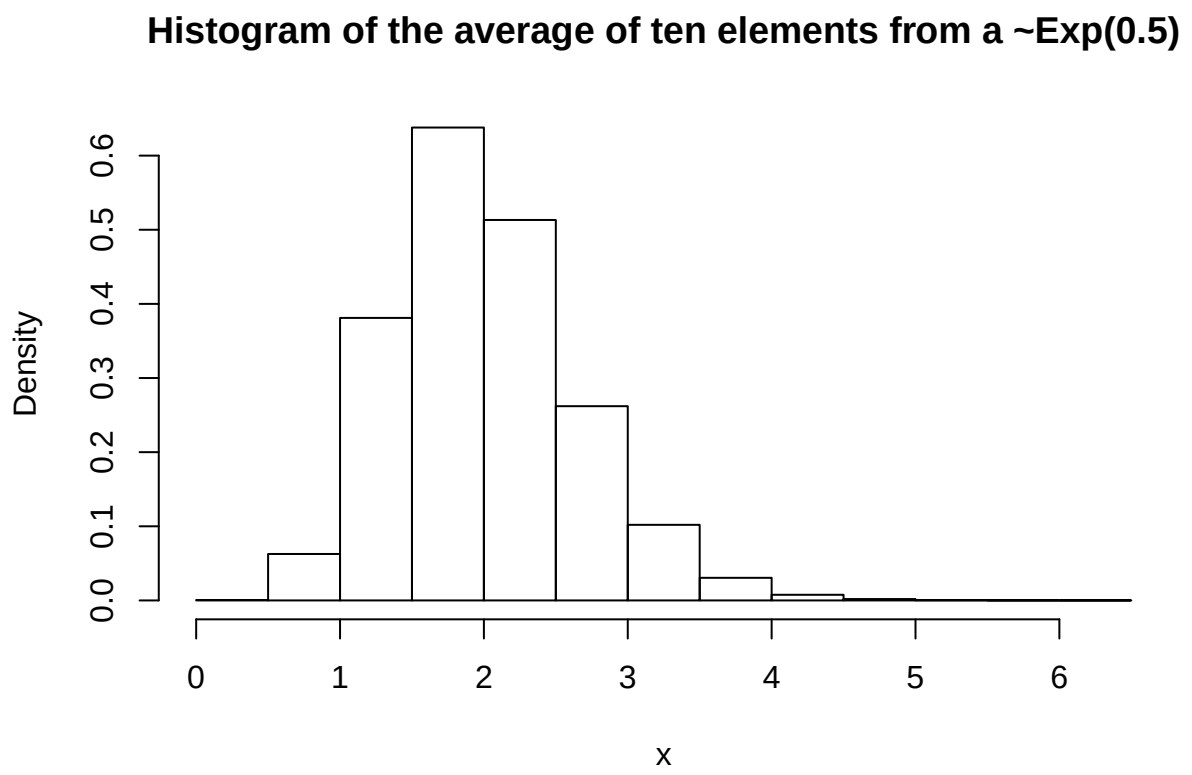

Figure 11: Histogram of 100,000 random means of ten elements from a ~Exp(0.5) distribution

By averaging ten elements, we can see in figure 11 that the distribution seems to be getting closer to a normal distribution centered around 2.

1.2.6

```
hist(myRexpMeans$Means30, main="Histogram of the average of thirty elements from a ~Exp(0.5)",
     xlab="x", freq = F)
```

In figure 12, we see the histogram when we take the average of 30 elements from the original negative exponential population shown in figure 7. The distribution is now very different, and looks like a normal distribution with mean 2. The tail on the right is still slightly more spread than the one on the left.

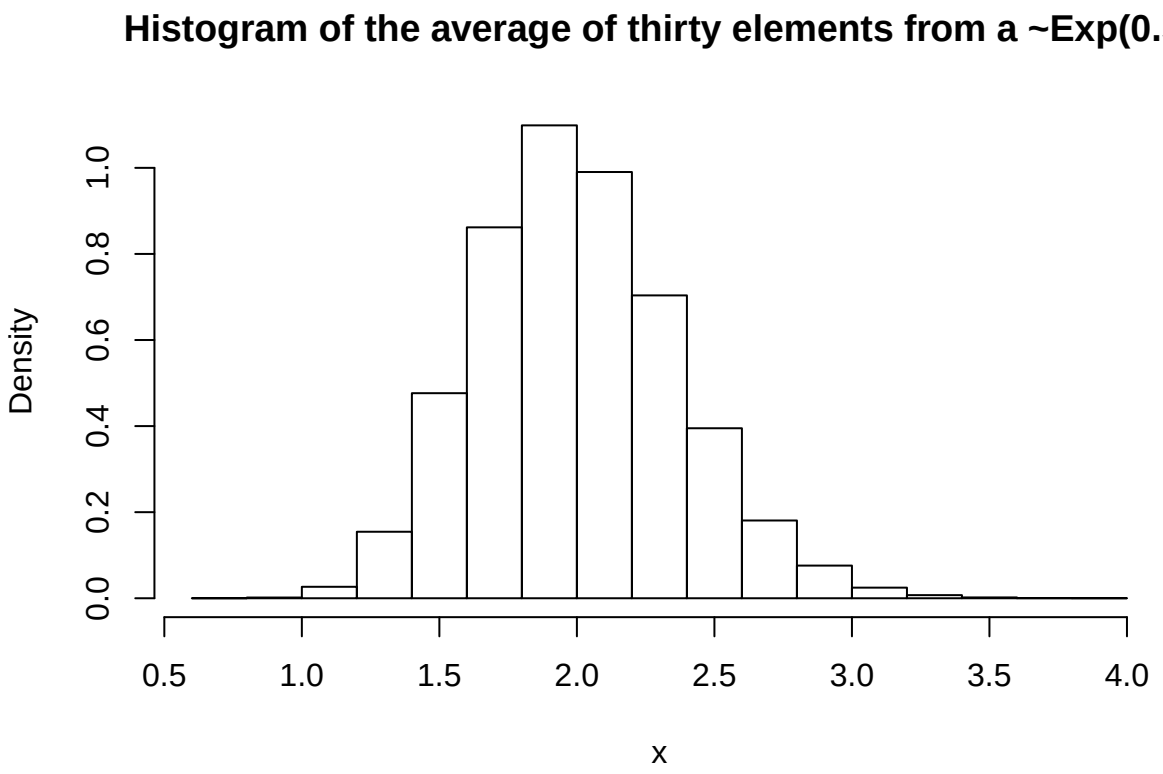

Figure 12: Histogram of 100,000 random means of thirty elements from a ~Exp(0.5) distribution

1.3

1.3.1

```
myBdist <- c(rnorm(5000000, -3, 1), rnorm(5000000, 3, 1))
hist(myBdist, main="Bimodal histogram of ~N(-3, 1) and ~N(3, 1)", xlab="x", freq = F)
```

We can see from the histogram in figure 13 that the distribution is bimodal. In fact, we have random variables from two distinct normal distributions, one centered around -3 and the other around 3. Both variances are equal to 1.

1.3.2

```
myBdistDataFrame <- data.frame(replicate(30, sample(myBdist, 100000)))
myBdistMeans <- data.frame("Means5"= apply(myBdistDataFrame[,1:5], 1, mean),
                           "Means10"= apply(myBdistDataFrame[,1:10], 1, mean),
                           "Means20"= apply(myBdistDataFrame[,1:20], 1, mean),
                           "Means30"= apply(myBdistDataFrame[,1:30], 1, mean))
```

```
hist(myBdistMeans$Means5, main="Histogram of the average of five elements
     from a ~N(-3, 1) and a ~N(3, 1)", xlab="x", freq = F)
```

We can gather from the histogram in image 14 that the bimodality seen in figure 13 is lost. Indeed, this histogram has only one "bump", and has a lot of values between -2 and 2.

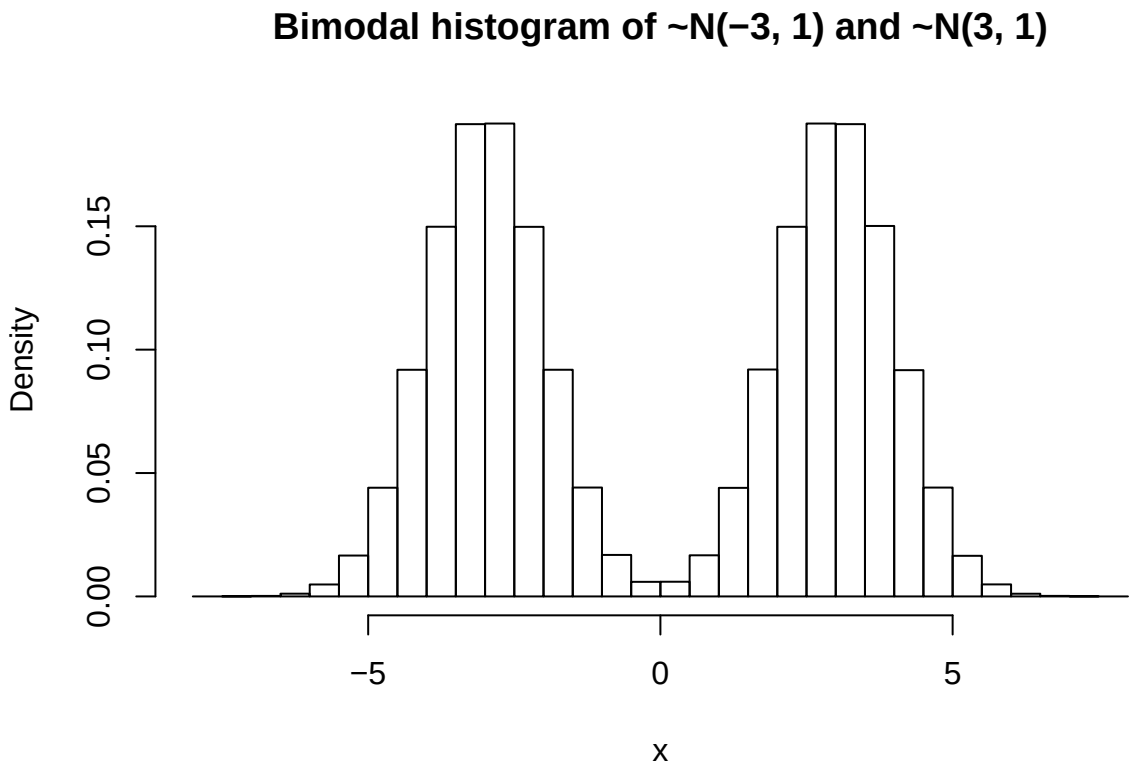

Figure 13: Histogram of 5,000,000 random variables from a ~N(-3, 1) and 5,000,000 random variables from a ~N(3, 1)

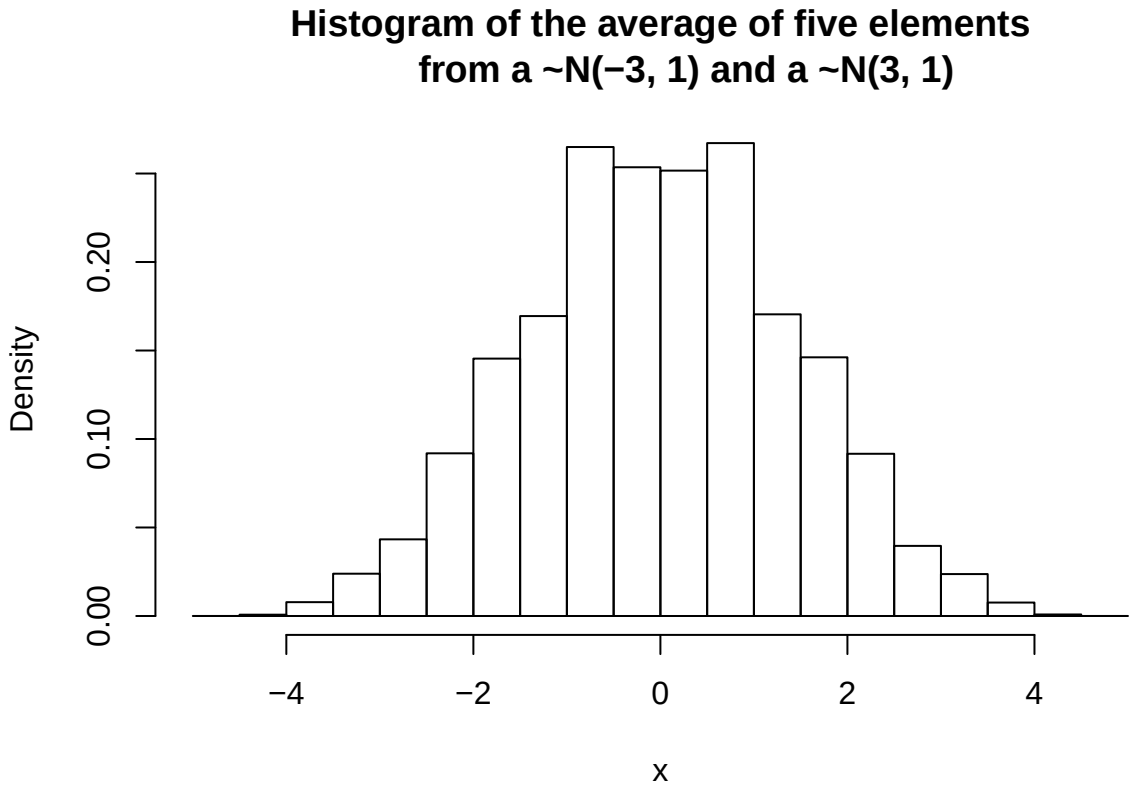

Figure 14: Histogram of 100,000 random means of five elements from a ~N(-3, 1) and a ~N(3, 1) distributions

1.3.3

```
hist(myBdistMeans$Means10, main="Histogram of the average of ten elements
     from a ~N(-3, 1) and a ~N(3, 1)", xlab="x", freq = F)
```

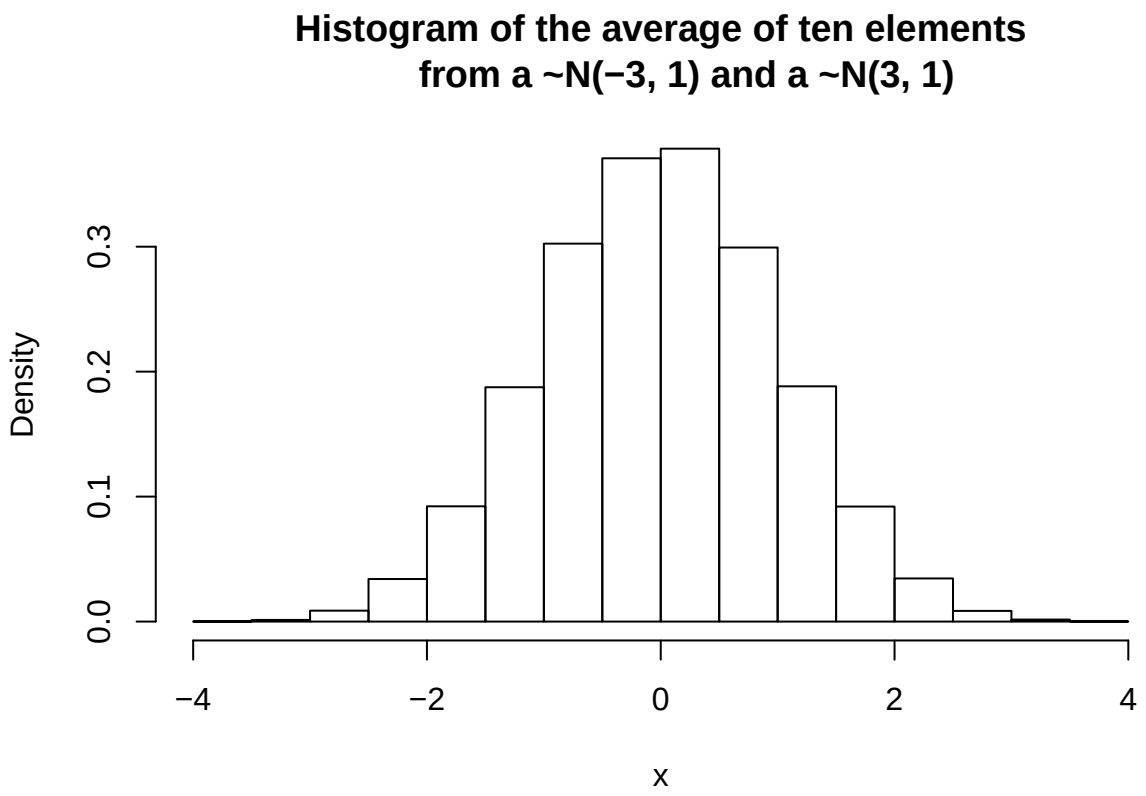

Figure 15: Histogram of 100,000 random means of ten elements from a ~N(-3, 1) and a ~N(3, 1) distributions

The histogram in figure 15 is similar to the previous one (figure 14), but less spread out. It is starting to look like a normal distribution.

```
hist(myBdistMeans$Means20, main="Histogram of the average of twenty elements from
     a ~N(-3, 1) and a ~N(3, 1)", xlab="x", freq = F)
```

By averaging over 20 elements from the original pouplation, the histogram seen in figure 16 is looking a lot like a normal distribution centered around 0. The spread of the distribution is becoming smaller the more elements we average.

```
hist(myBdistMeans$Means30, main="Histogram of the average of thirty elements from
     a ~N(-3, 1) and a ~N(3, 1)", xlab="x", freq = F)
```

The histogram in figure 17 does not look like the original population distribution in figure 13. In fact, the two modes that were present at first have completely disappeared. In its place, we seems to have a single standard normal distribution, centered around the average of the original two means: $\frac{-3+3}{2} = 0$.

1.3.4

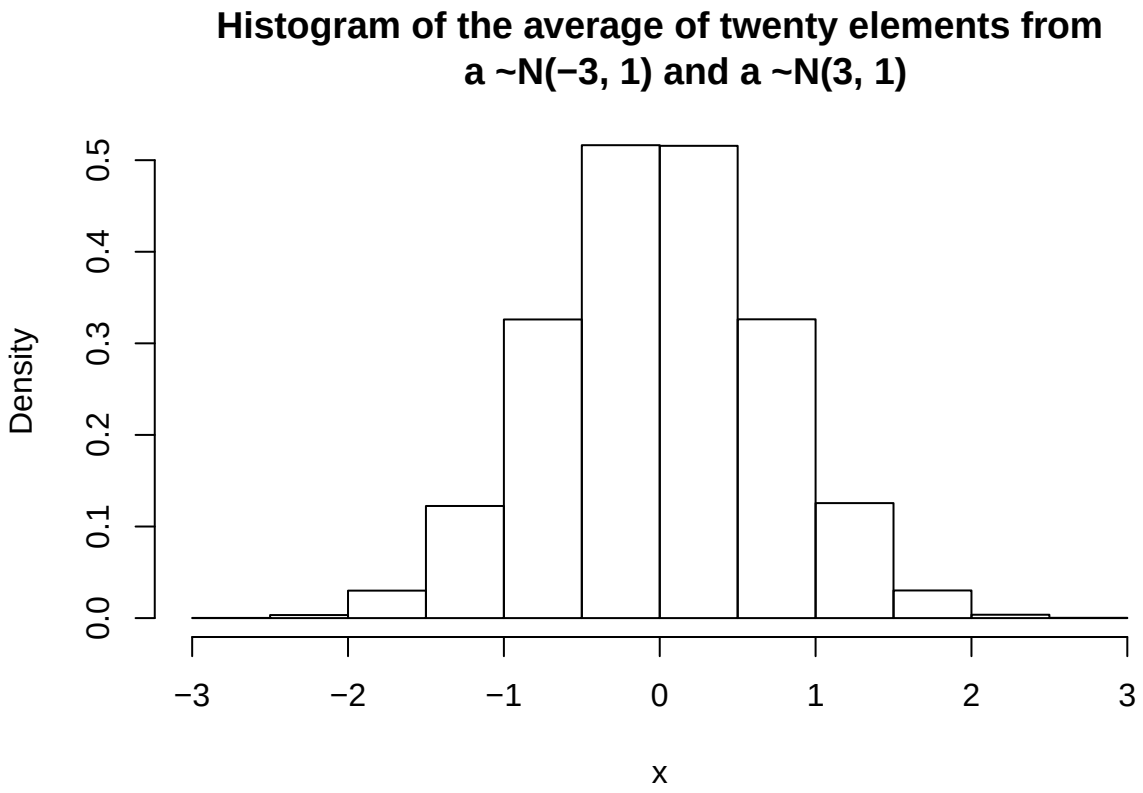

Figure 16: Histogram of 100,000 random means of twenty elements from a ~N(-3, 1) and a ~N(3, 1) distributions

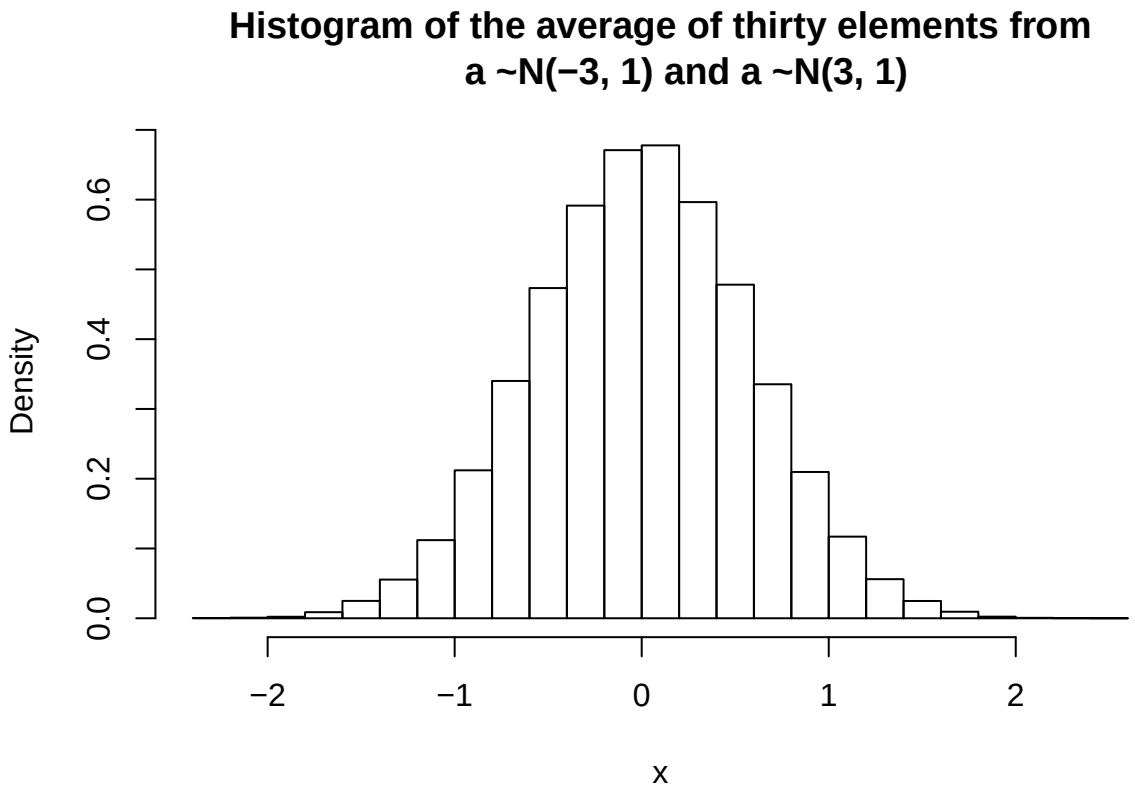

Figure 17: Histogram of 100,000 random means of thirty elements from a ~N(-3, 1) and a ~N(3, 1) distributions

Question 2

```
myData <- read.csv("hw2.csv", header = T, stringsAsFactors = F)
names(myData)
```

```
##  [1] "Project.Name"
##  [2] "Created.Date"
##  [3] "Project.Status"
##  [4] "Opportunity..Purchased.Thru"
##  [5] "Agreement.Type"
##  [6] "Install.Branch"
##  [7] "Opportunity..Utility.Company"
##  [8] "Opportunity..Jurisdiction..Jurisdiction.Name"
##  [9] "Proposal..System.Size.STC.DC"
## [10] "Service.Contract..Service.Contract.Event..Using.Build.Partner."
## [11] "Opportunity..Service.Panel.Upgrade"
## [12] "Opportunity..Reroof.under.Array"
## [13] "Opportunity..HOA."
## [14] "Opportunity..PE.Stamp.Required."
```

```
str(myData, strict.width = "w")
```

```
## 'data.frame':    49880 obs. of  14 variables:
```

```
## $ Project.Name : chr "PR-1631525512" "PR-1727346069" "PR-1321701825"
##    "PR-1416881773" ...
## $ Created.Date : chr "11/19/15" "5/31/15" "1/25/16" "11/19/15" ...
## $ Project.Status : chr "Open" "Cancelled" "Open" "Open" ...
## $ Opportunity..Purchased.Thru : chr "Costco" "" "" "Costco" ...
## $ Agreement.Type : chr "Customer Owned - Full Upfront" "Custom PPA Fixed"
##    "Custom PPA Fixed" "Customer Owned - Bank Financed" ...
## $ Install.Branch : chr "Sacramento" "Las Vegas" "MD - Columbia" "Inland
##    Empire" ...
## $ Opportunity..Utility.Company : chr "PG&E" "NV Energy South" "BG&E" "SCE"
##    ...
## $ Opportunity..Jurisdiction..Jurisdiction.Name : chr "CA-CITY CHICO"
##    "NV-COUNTY CLARK" "MD-COUNTY BALTIMORE" "CA-CITY NORCO" ...
## $ Proposal..System.Size.STC.DC : num 5.57 5.72 8.48 5.83 7.54 ...
## $ Service.Contract..Service.Contract.Event..Using.Build.Partner.: int 0 0
##    1 0 0 0 0 0 0 1 ...
## $ Opportunity..Service.Panel.Upgrade : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Opportunity..Reroof.under.Array : int 0 0 0 0 0 0 0 0 0 0 ...
## $ Opportunity..HOA.  : chr "No" "Yes" "Yes" "No" ...
## $ Opportunity..PE.Stamp.Required.  : int 0 0 1 1 0 1 0 0 0 1 ...
```

number of rows = 49880

number of cols = 14

```r
myData$Service.Contract..Service.Contract.Event..Using.Build.Partner. <- as.logical(myData$Service.Cont
myData$Opportunity..Service.Panel.Upgrade <- as.logical(myData$Opportunity..Service.Panel.Upgrade)
myData$Opportunity..Reroof.under.Array <- as.logical(myData$Opportunity..Reroof.under.Array)
myData$Opportunity..PE.Stamp.Required. <- as.logical(myData$Opportunity..PE.Stamp.Required.)

myData$Opportunity..HOA.[myData$Opportunity..HOA. == "0"] <- "No"
myData$Opportunity..HOA.[myData$Opportunity..HOA. == "1"] <- "Yes"
myData$Opportunity..HOA. <- factor(myData$Opportunity..HOA., levels = unique(myData$Opportunity..HOA.))

myData$Project.Status <- factor(myData$Project.Status, levels = unique(myData$Project.Status))

myData$Opportunity..Purchased.Thru[myData$Opportunity..Purchased.Thru == "Costco (Dept 44)"] <- "Costco
myData$Opportunity..Purchased.Thru <- factor(myData$Opportunity..Purchased.Thru,
    levels = unique(myData$Opportunity..Purchased.Thru))
myData$Agreement.Type <- factor(myData$Agreement.Type, levels = unique(myData$Agreement.Type))
myData$Opportunity..Utility.Company <- factor(myData$Opportunity..Utility.Company,
    levels = unique(myData$Opportunity..Utility.Company))

# unique(myData$Install.Branch) # uniformize the data (STATE - City)
# unique(myData$Install.Branch) # uniformize the data (STATE - City)
# unique(myData$Install.Branch) # uniformize the data (regular expressions?)
```