MATH382L R Lab 11 Central Limit Theorem. Functions of random variables.

Jennifer Minnich
April 14, 2021

1 Problem 1

- a. Find (theoretically) E(Xi) and V (Xi).

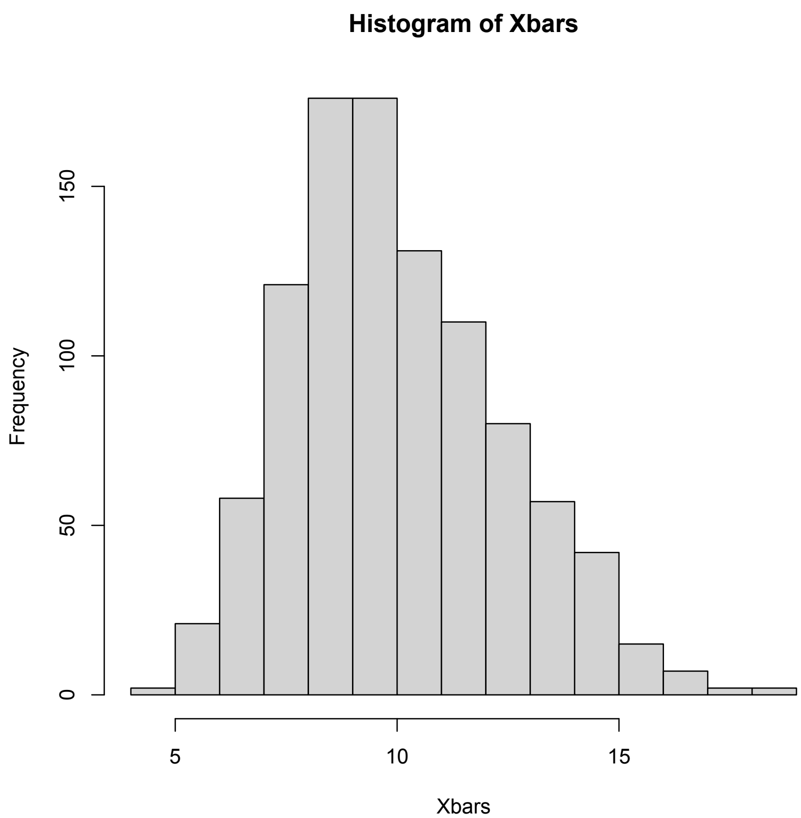
```
> #Problem 1a
> alpha=5
> beta=2
> EX = alpha*beta
> EX
[1] 10
> VX = alpha*(beta^2)
> VX
[1] 20
```

- b. Look at n = 4. i. Create a sampling distribution for the sample mean using for loop1:

```
> #1b
> N = 1000 # number of samples generated
> n = 4 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rgamma(n, alpha, 1/beta)
+ Xbars[i] = mean(Xsamp)
+ }
> Xbars
   [1]  9.464965  6.844450 11.858892  7.974946 13.542646
   [6] 12.130340 11.643689 14.160040  7.829668 11.906431
  [11]  8.205398 12.192158 13.599372 12.104555  9.461341
  [16]  7.167656  9.357683 13.810890 13.023047 12.126641
  [21]  8.437218  8.021301  7.979074 10.160403 12.238336
```

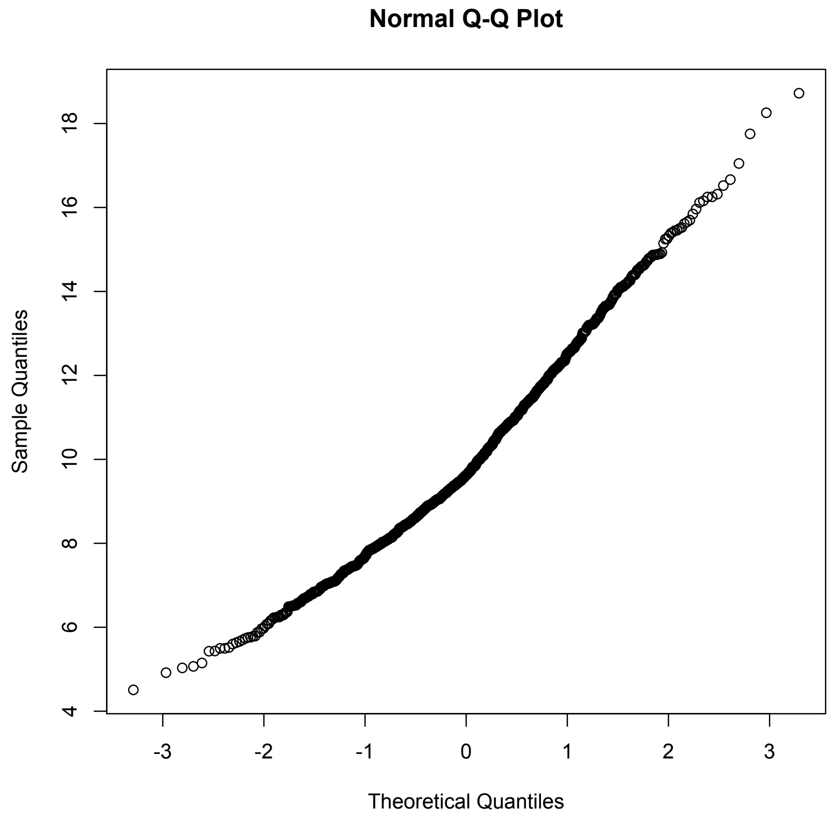
ii. Create a histogram and calculate the descriptive statistics for Xbars.

```
> #1c
> hist(Xbars)
> mean(Xbars)
[1] 10.00814
> sd(Xbars)
[1] 2.371886
> quantile(Xbars)
       0%       25%       50%       75%      100%
 4.508804  8.283571  9.618961 11.528590 18.721869
```



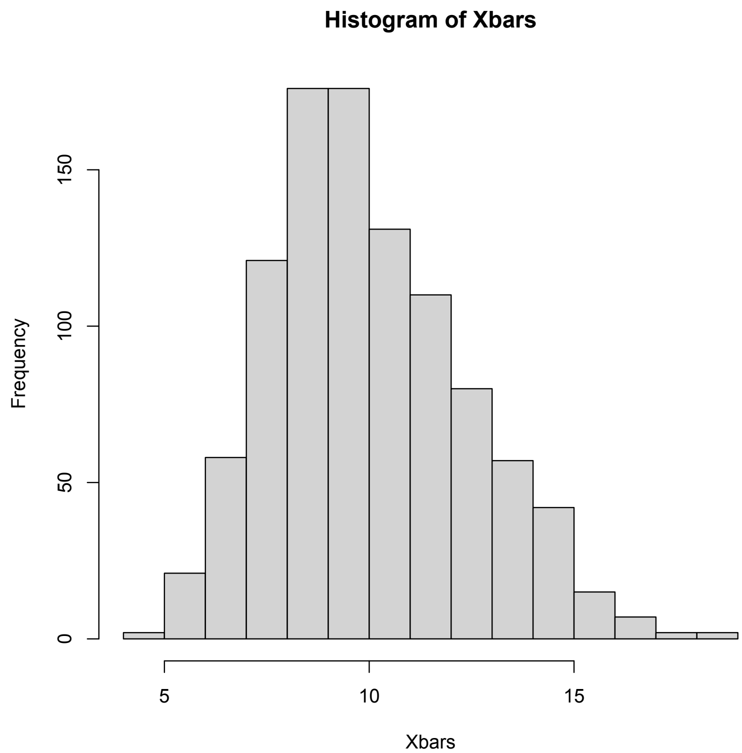
iii. Is the histogram approximately normal? Make a normal q-q plot.

Looking at our histogram, the graph has a wide shaped curve that spreads around the mean but skews at the right. This approximation is leaning towards a normal distribution, however, a larger sample size is necessary to derive a conclusion. Our qq plot confirms our assumption of a normal distribution, since our points are creating a straight line.



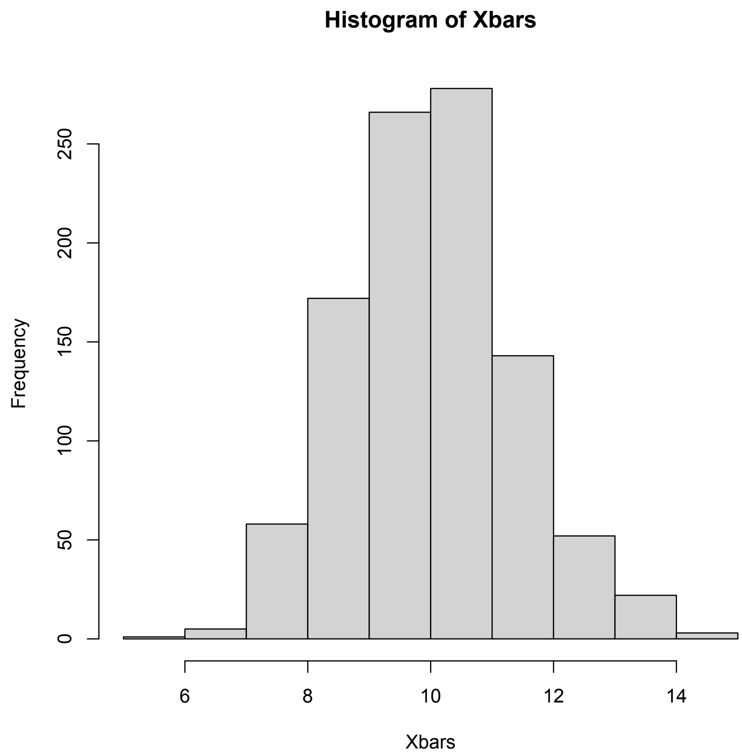
iv. What is (theoretically) E(X) and V (X)? How do these values compare to the descriptive statistics in part ii?.

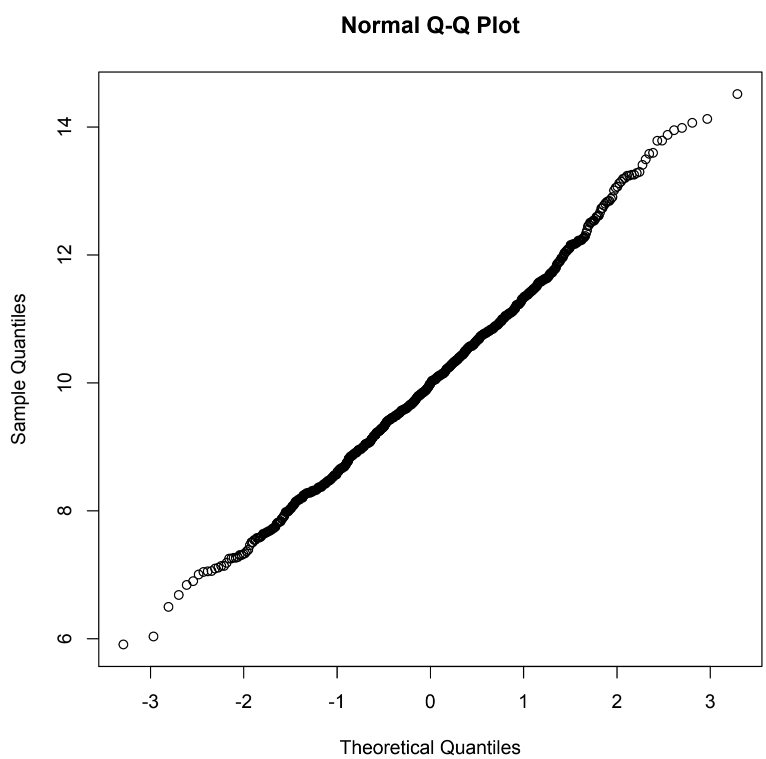
Our E(X) = 10 and our V(X) = 20. The statistic of note is the standard deviation, at 2.37 which reflects the data spread out over a wider range.



- c. Repeat part (b) with n = 10. Now does the histogram look approximately normal? If not, increase n and repeat.

```
> #1c
> N = 1000 # number of samples generated
> n = 10 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rgamma(n, alpha, 1/beta)
+ Xbars[i] = mean(Xsamp)
+ }
> hist(Xbars)
> mean(Xbars)
[1] 10.00245
> sd(Xbars)
[1] 1.366829
> quantile(Xbars)
       0%       25%       50%       75%      100%
 5.910937  9.057614  9.986749 10.853608 14.516070
```





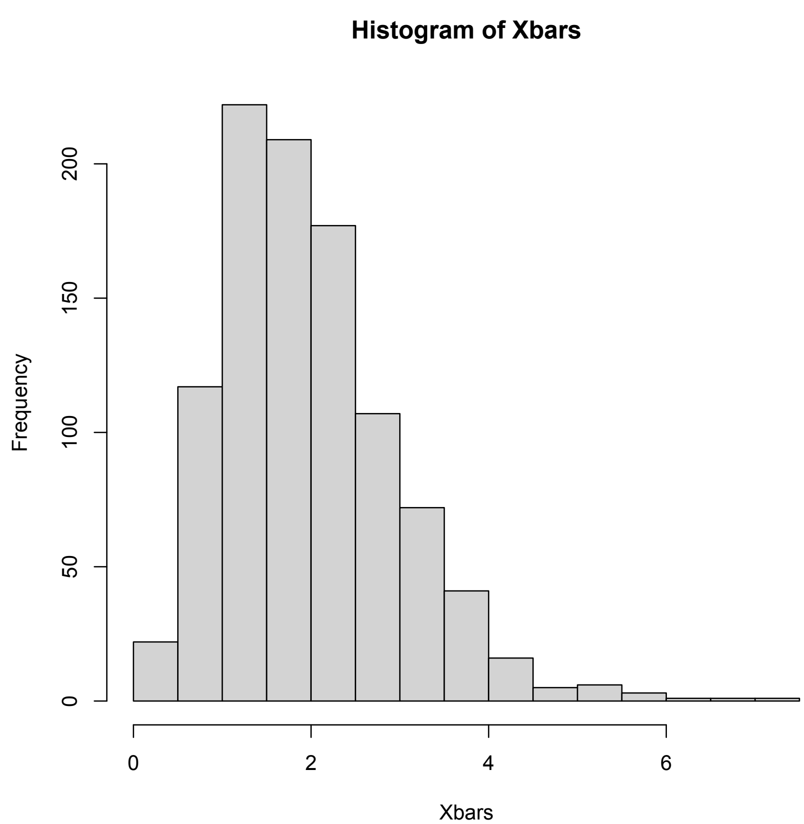
The histogram certainly looks closer to a defined normal distribution. The deviation decreased to 1.36 which concentrates more of the data about the mean, has less of a spread , and doesn't skew to one side of the other.

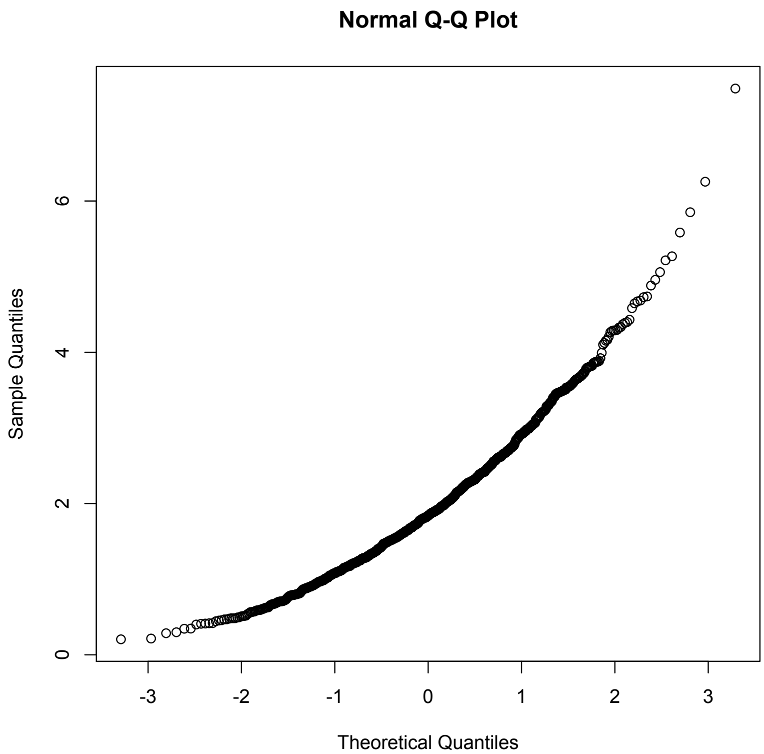
The qq plot also looks more linear in shape with no curve or dips.

2 Problem 2

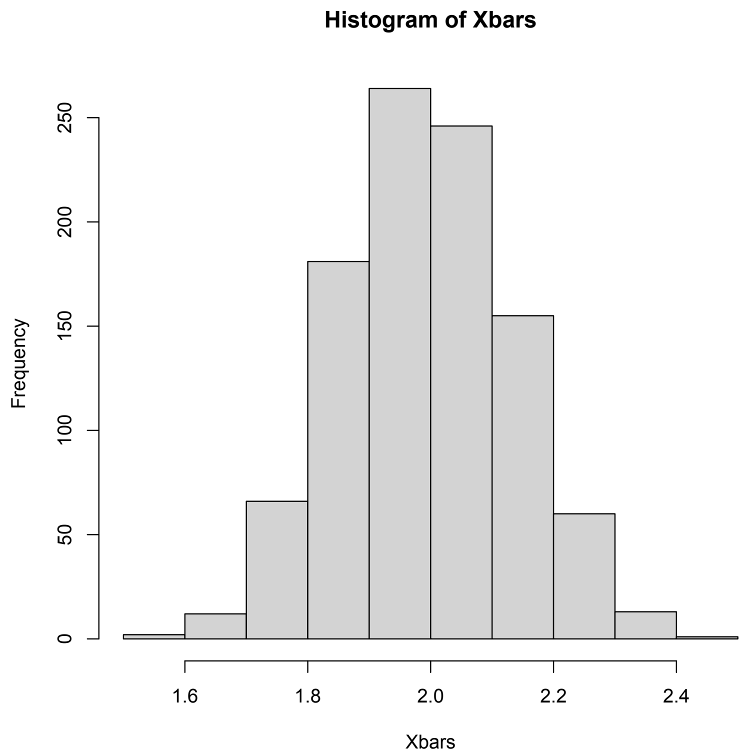
- a. Try another probability distribution; say an exponential with mean 2. Repeat the process above and then increase the sample size until your distribution of sample means looks approximately normal.

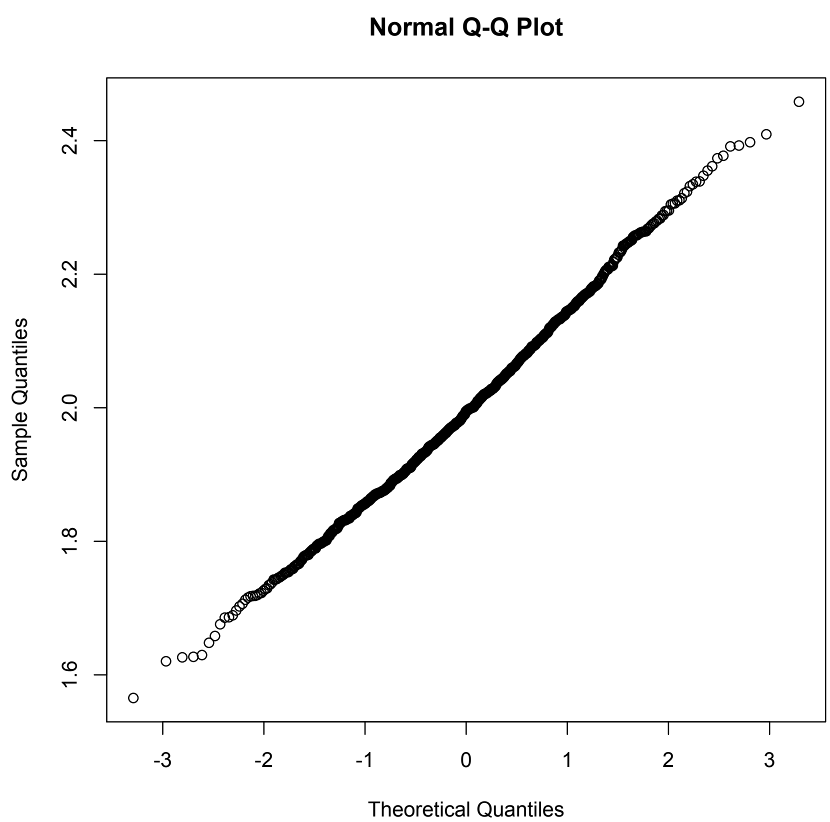
```
> #Problem 2
> beta=2
> N = 1000 # number of samples generated
> n = 4 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rexp(n, 1/beta)
+ Xbars[i] = mean(Xsamp)
+ }
> #Xbars
> hist(Xbars)
> mean(Xbars)
[1] 1.975453
> sd(Xbars)
[1] 0.9898905
> quantile(Xbars)
       0%       25%       50%       75%      100%
0.2122962 1.2292108 1.8097758 2.5203598 7.1081187
```






```
> #Problem 2
> beta=2
> N = 1000 # number of samples generated
> n = 200 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rexp(n, 1/beta)
+ Xbars[i] = mean(Xsamp)
+ }
> #Xbars
> hist(Xbars)
> mean(Xbars)
[1] 1.997937
> sd(Xbars)
[1] 0.1437508
> quantile(Xbars)
      0%      25%      50%      75%     100%
1.565348 1.896075 1.994922 2.093270 2.458292
>
```


Histogram of Xbars
Frequency
0
50
100
150
200
250
1.6
1.8
2.0
2.2
2.4
Xbars


Normal Q-Q Plot
Sample Quantiles
1.6
1.8
2.0
2.2
2.4
-3
-2
-1
0
1
2
3
Theoretical Quantiles

- b. Looking at your results in problems 1-2, do you need the same sample sizes in each case to get a distribution that is approximately normal? What features of the initial distribution affect n needed to reach approximate normality?

  The exponential distribution needed a larger sample size in order to get an approximately normal distribution (n=200), the gamma distribution required a relatively small sample size (n=10).

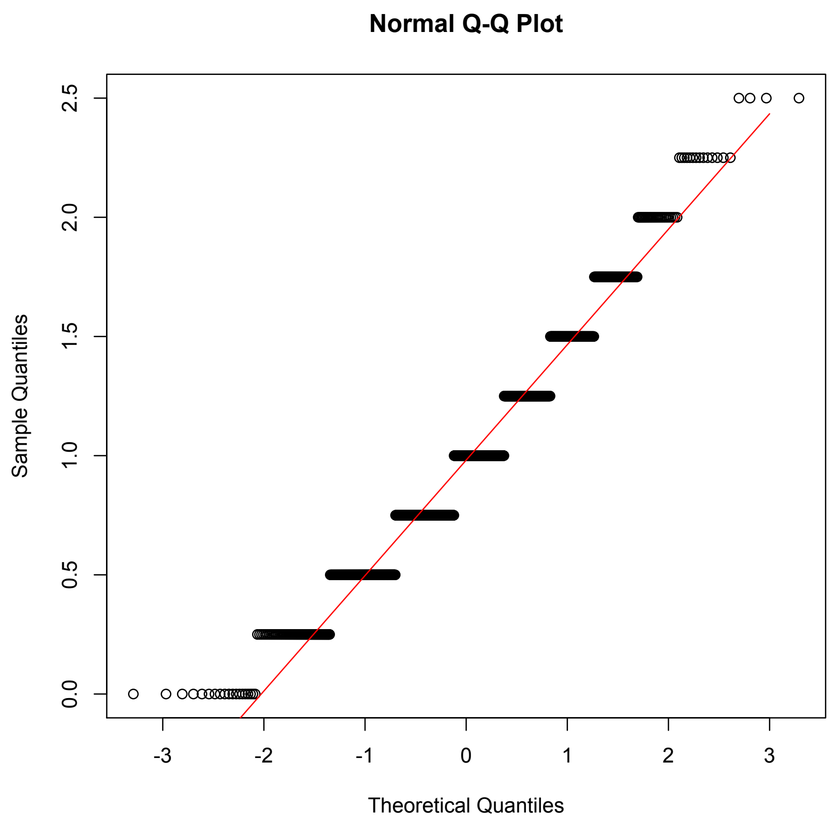
3 Problem 3

- a. What is the E(p) and V (p)?

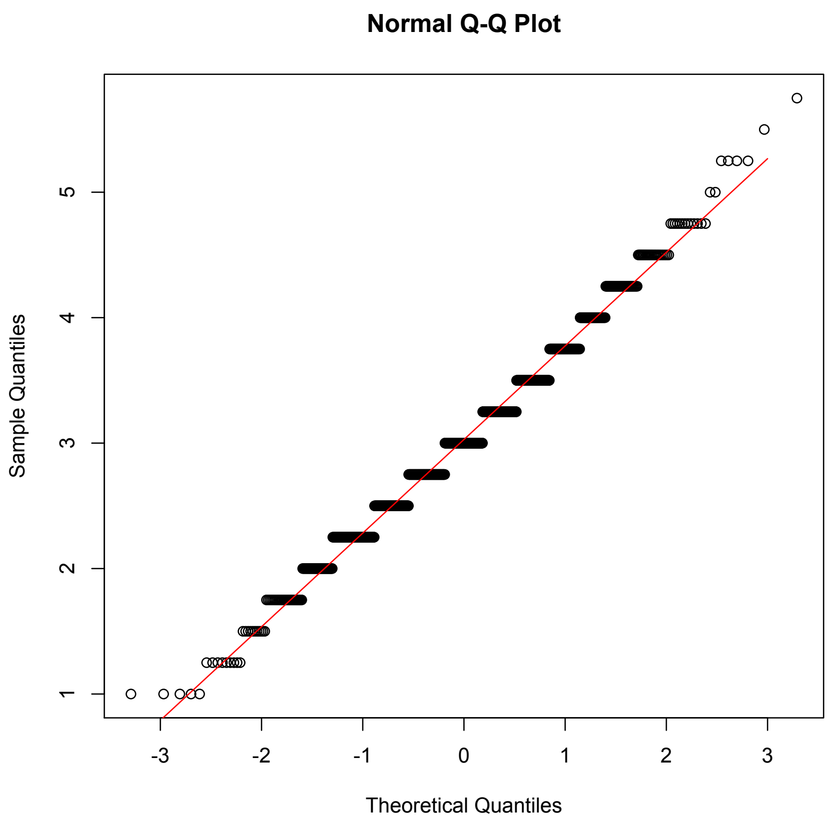
  #Problem 3a
  E(p hat)= E(X/n) = np/n = p
  V(p hat)= V(X/n) = npq/n2 = pq/n

- b. How large a sample size do we need for the sampling distribution to be approximately normal? Try several values of p, say 0.1, 0.3, and 0.5. Relate your findings to the situation in Problems 1, 2.

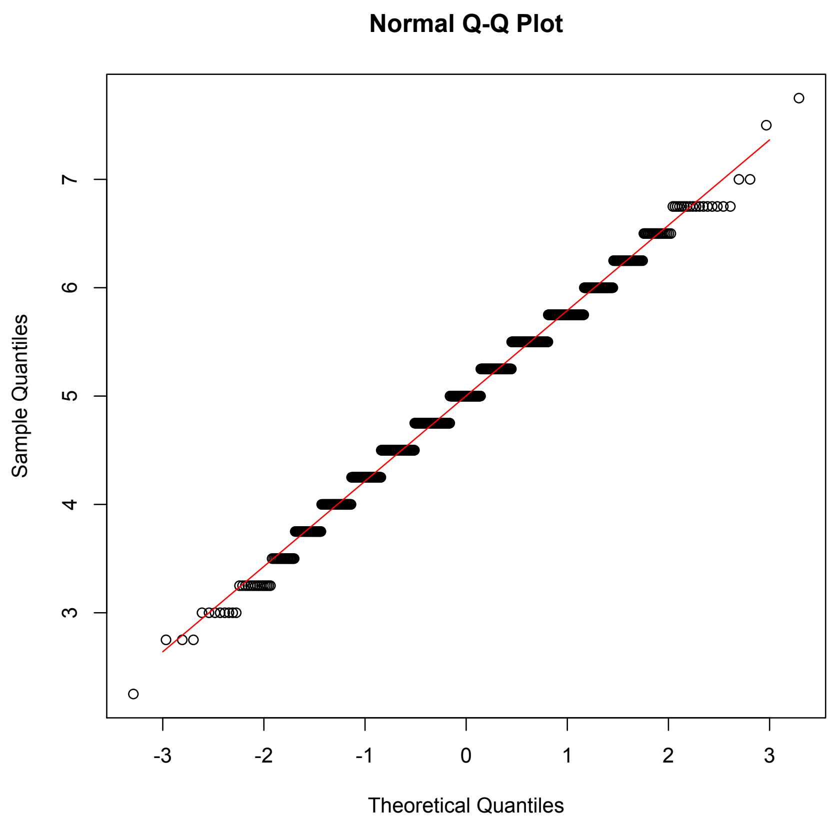
```
> #Problem 3b
> p = .1
> N = 1000 # number of samples generated
> n = 4 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rbinom(n, 10, prob=p)
+ Xbars[i] = mean(Xsamp)
+ }
> qqnorm(Xbars)
> lines(seq(-3,3,0.1), (seq(-3,3,0.1)*sd(Xbars))+mean(Xbars),
col='red')
```



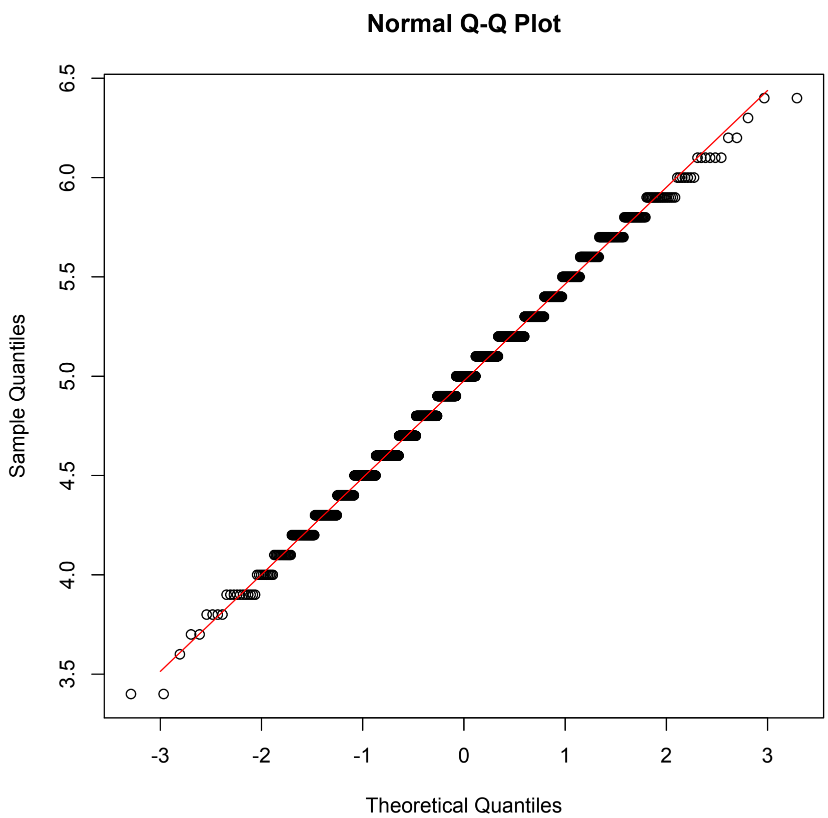

```
> p = .3
> N = 1000 # number of samples generated
> n = 4 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rbinom(n, 10, prob=p)
+ Xbars[i] = mean(Xsamp)
+ }
> qqnorm(Xbars)
> lines(seq(-3,3,0.1), (seq(-3,3,0.1)*sd(Xbars))+mean(Xbars),
col='red')
```




```
> p = .5
> N = 1000 # number of samples generated
> n = 4 # sample size
> Xbars = numeric(N)
> for (i in 1:N)
+ {
+ Xsamp = rbinom(n, 10, prob=p)
+ Xbars[i] = mean(Xsamp)
+ }
> qqnorm(Xbars)
> lines(seq(-3,3,0.1), (seq(-3,3,0.1)*sd(Xbars))+mean(Xbars),
col='red')
```



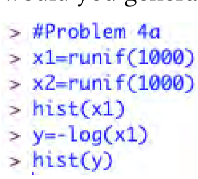

```r
p = .5
N = 1000 # number of samples generated
n = 10 # sample size
Xbars = numeric(N)
for (i in 1:N)
{
Xsamp = rbinom(n, 10, prob=p)
Xbars[i] = mean(Xsamp)
}
qqnorm(Xbars)
lines(seq(-3,3,0.1), (seq(-3,3,0.1)*sd(Xbars))+mean(Xbars), col='red')
```



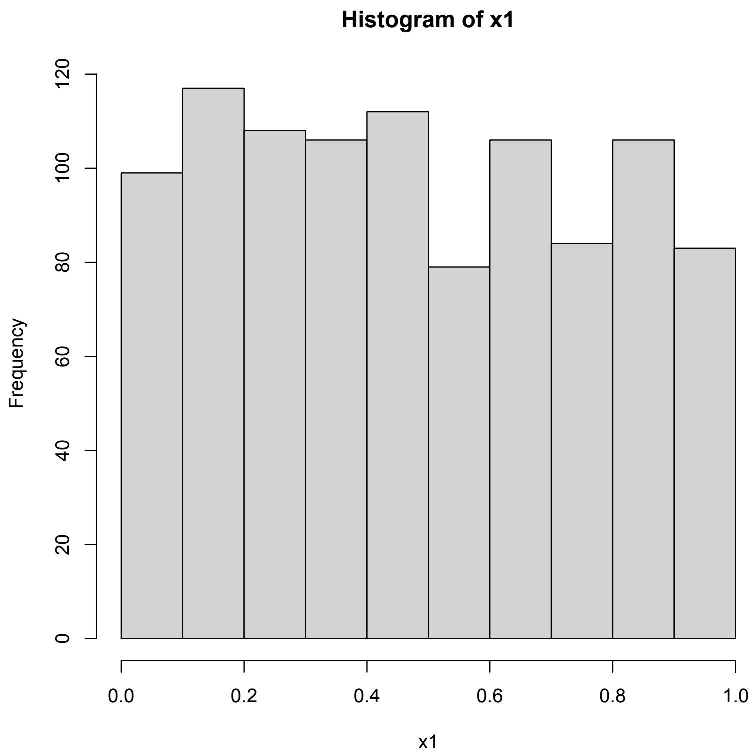
As the sample size increases from 4 to 10 and the p increases from .1 to .5, the distribution gets closer to approximately normal. Looking at our last qq plot, the line is almost straight as we hypothesize for normal distribution.

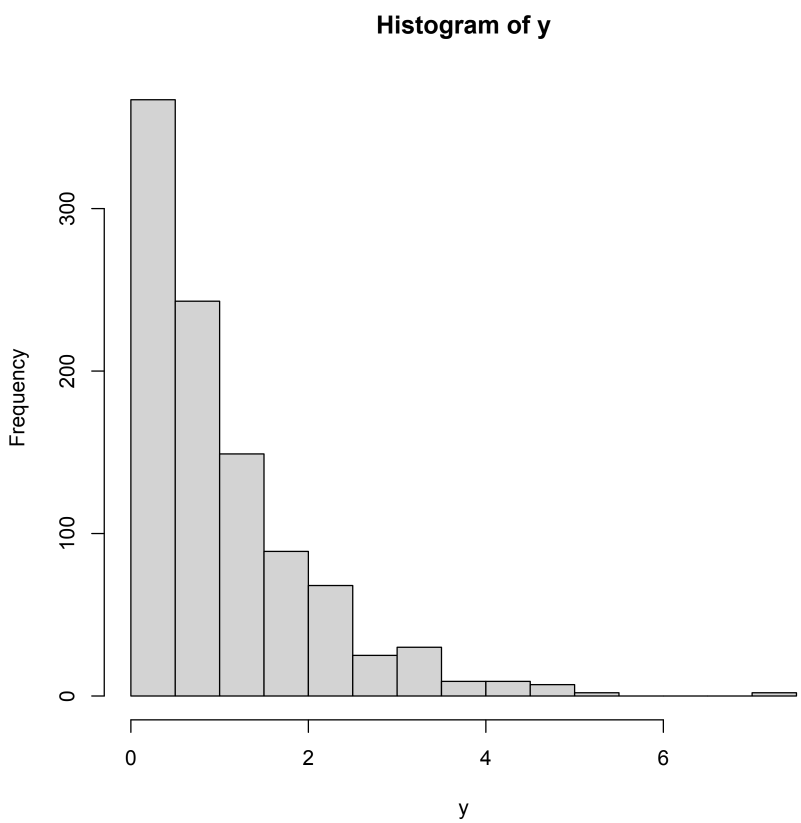
4 Problem 4

- a.Obtain a sample from Exponential(1) distribution. Make a histogram to make sure it "looks right". How would you generate Exponential(64) distribution?



```
> #Problem 4a
> x1=runif(1000)
> x2=runif(1000)
> hist(x1)
> y=-log(x1)
> hist(y)
```

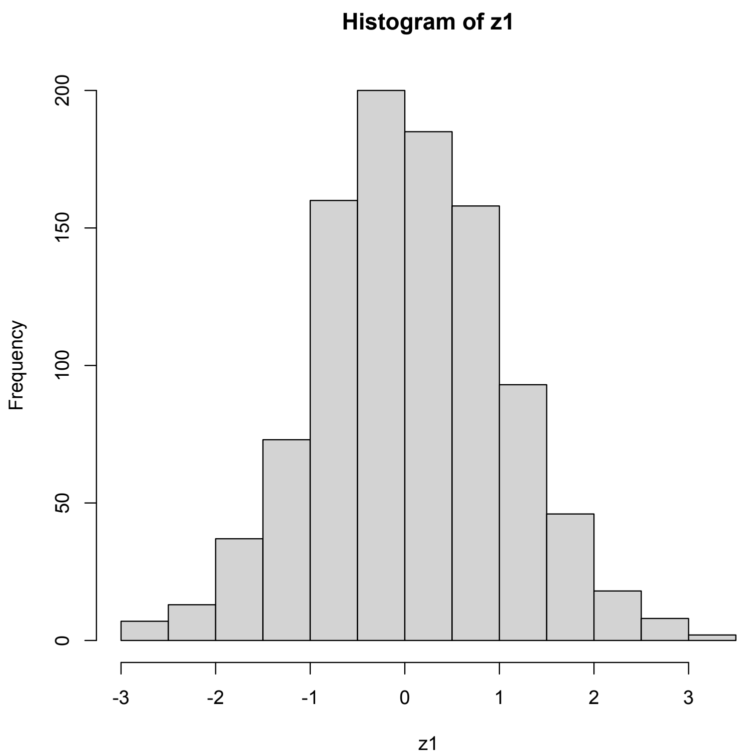

Histogram of x1
Frequency
0
20
40
60
80
100
120
0.0
0.2
0.4
0.6
0.8
1.0
x1

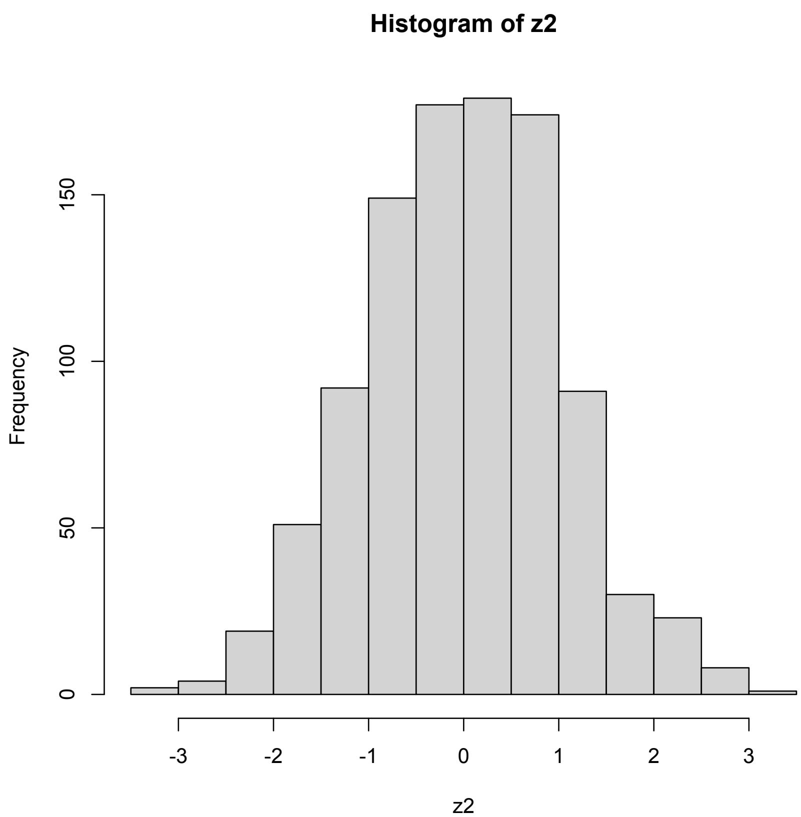

Histogram of y
Frequency
0
100
200
300
0
2
4
6
y

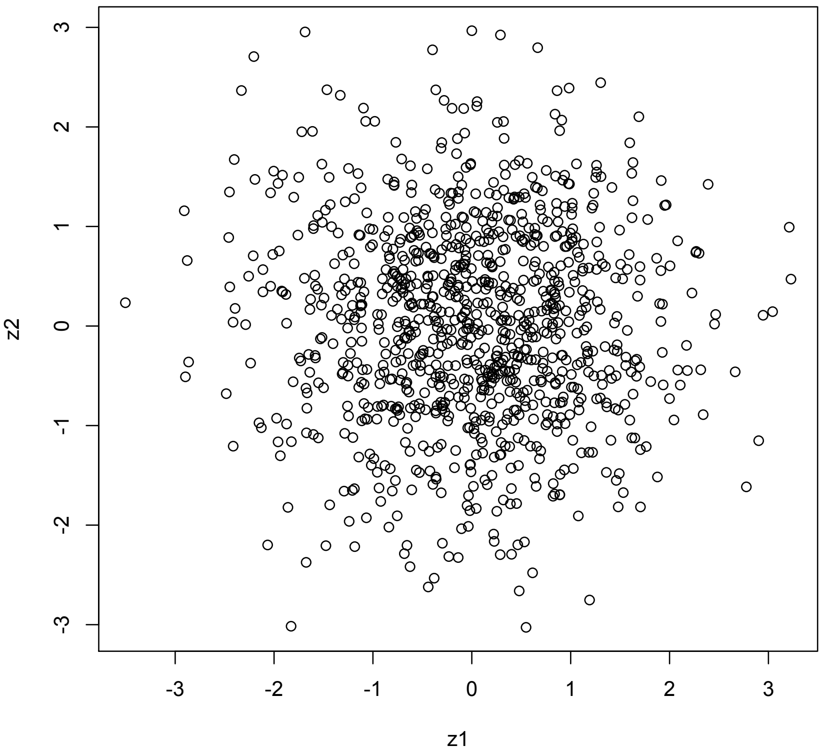
Yes, the histogram of y definitely looks exponential.

- b.two samples (columns) of standard Normal distribution. Make histograms and a scatterplot to check independence.

```
> #4b
> x3=runif(1000,0,2*3.1416)
> z1=sqrt(2*y)*cos(x3)
> z2=sqrt(2*y)*sin(x3)
> hist(z1)
> hist(z2)
> plot(z1,z2)
```

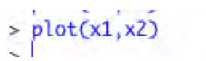


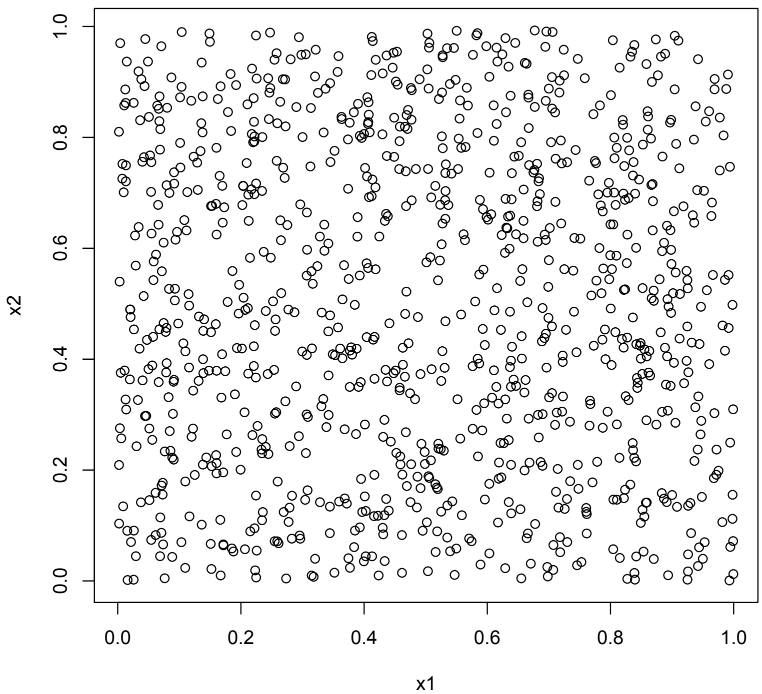

Histogram of z2
Frequency
0
50
100
150
-3
-2
-1
0
1
2
3
z2



We can determine independence based on the scatterplot's normal distribution.

- c. Make a scatterplot of the two original Uniform [0,1] RV's, compare its shape with the plot of Normals from part (b).





The distribution is normal and there is no pattern seen in the plot.