STATISTICAL METHODS FOR DATA SCIENCE
MINI PROJECT #1

NAME: ADRITA DUTTA
NET-ID: axd172930

SPRING 2018

SECTION 1:

Q1.Consider a discrete random variable X that takes 4 values-1, 2, 3 and 4 with respective probabilities 1/2, 1/8, 1/8, and 1/4.

(a) Compute E(X), var(X) and P(X # 2) analytically, i.e., using their formulas.

Answer:
1) X= x_1, x_2, x_3, x_4= 1, 2, 3, 4
2) P=p_1, p_2, p_3, p_4=1/2, 1/8, 1/8, 1/4

=> $E(X) = \mu$

$$= x_1p_1+x_2p_2+x_3p_3+ \ldots +x_np_n.$$

$$= \left(1\left(\frac{1}{2}\right)\right)+\left(2\left(\frac{1}{8}\right)\right)+\left(3\left(\frac{1}{8}\right)\right)+\left(4\left(\frac{1}{4}\right)\right)$$

$$= 0.5+0.25+0.37+1$$

$E(X) = 2.125$

3)

x_i	p_i	$(x_i-\mu)^2$
1	1/2	$(1-2.125)^2=1.265$
2	1/8	$(2-2.125)^2=0.0156$
3	1/8	$(3-2.125)^2=0.7656$
4	1/4	$(4-2.125)^2=3.5156$

=> $var(X) =\Sigma\, \sigma^2 p_i$ [1<= i <= 4]

$$=\Sigma\ (Xi-\ \mu)\otimes Pi$$

$$= \left(1.265\left(\frac{1}{2}\right)\right)+\left(0.0156\left(\frac{1}{8}\right)\right)+\left(0.7656\left(\frac{1}{8}\right)\right)+\left(3.5156\left(\frac{1}{4}\right)\right)$$

$$= 0.63+0.001+0.09+0.87$$

var(X) =1.60905

=>P(X<=2)= P(1)+P(2)

$$= p_1+p_2$$

$$= \left(\frac{1}{2}\right)+\left(\frac{1}{8}\right)$$

P(X<=2) =0.625

(b) Explain how you would simulate a draw from the distribution of X.

Answer:

Knowing discrete random variable X takes values x_1, x_2, x_3, x_4 with probabilities p1, p2, p3, p4

pi = P {X =xi}

1)Divide the interval A=[1,4] into subintervals [A_1,A_2,A_3,A_4]

A_1=[0,p_1)

A_2=[p1, p1+p2)

A_3=[p_1+p_2, p_1+p_2+p_3)

A_4=[p_1+p_2+p_3, p_1+p_2+p_3+p_4)

2)obtain standard uniform Random Variable U

3) If $U \in A$, let $X \in Xi$

P{X=x_i}= P{ $U \in Ai$ }=p_i

<u>(c) Approximate E(X), var(X) and P(X # 2) using Monte Carlo simulation with 1,000 draws 5 times. Summarize the results in a table.</u>

=> E(X) using Monte Carlo= E(X_bar)

$$= \sum E\frac{(Xi)}{N} \qquad [1<= i <= 4]$$

$$= \frac{1}{N} \otimes N\mu$$

$$= \mu$$

=> Var(X) using Monte Carlo = Var(X_bar)

$$= \sum Var\frac{(Xi)}{N^2}$$

$$= \frac{1}{N^2} \otimes N \otimes \sigma$$

$$= \frac{\sigma^2}{N}$$

=> P(X<=2) using Monte Carlo = $\frac{(\sum Pi(1)+Pi(2))}{N}$

Results in table:

1000 draws 5 times(Code for obtaining following results in Section 2)

Repitation	E(X)	Var(X)	P(X<=2)
1	2.495858	1.374329	0.4193617
2	2.490398	1.202729	0.4490126
3	2.456381	1.292734	0.4426159
4	2.53589	1.414128	0.4195495
5	2.518635	1.270768	0.3914974

<u>(d) Repeat (c) with 5,000 and 10,000 draws.</u>

5000 draws 5 times(Code for obtaining following results in Section 2)

Repitation	E(X)	Var(X)	P(X<=2)
1	2.486245	1.252768	0.4123660
2	2.512163	1.223974	0.4106074
3	2.518696	1.233815	0.4089927
4	2.487804	1.276931	0.4236489
5	2.501934	1.254684	0.4124114

10000 draws 5 times(Code for obtaining following results in Section 2)

Repitation	E(X)	Var(X)	P(X<=2)
1.	2.509946	1.232619	0.4189146
2.	2.489397	1.235854	0.4190509
3.	2.104793	1.714490	0.4823175
4.	2.147993	1.255690	0.4112808
5.	2.485007	1.693625	0.4433729

(e) Compare you results in (a), (c) and (d). Explain, with justication, what you observe.

Answer:
According to the result of a, b and c
We can see that as the value of n increases the value of E(X), Var(X) and P(X<=2) becomes closer to the value at normal distribution.
We can see that values when n=10000 is closer to the values in a than those of n=5000 or n=1000 and with the same analogy values of n=5000 are closer to a than those of n=1000.

Thus, we can say that the CLT(Central Limit Theorem) is applicable in this case.

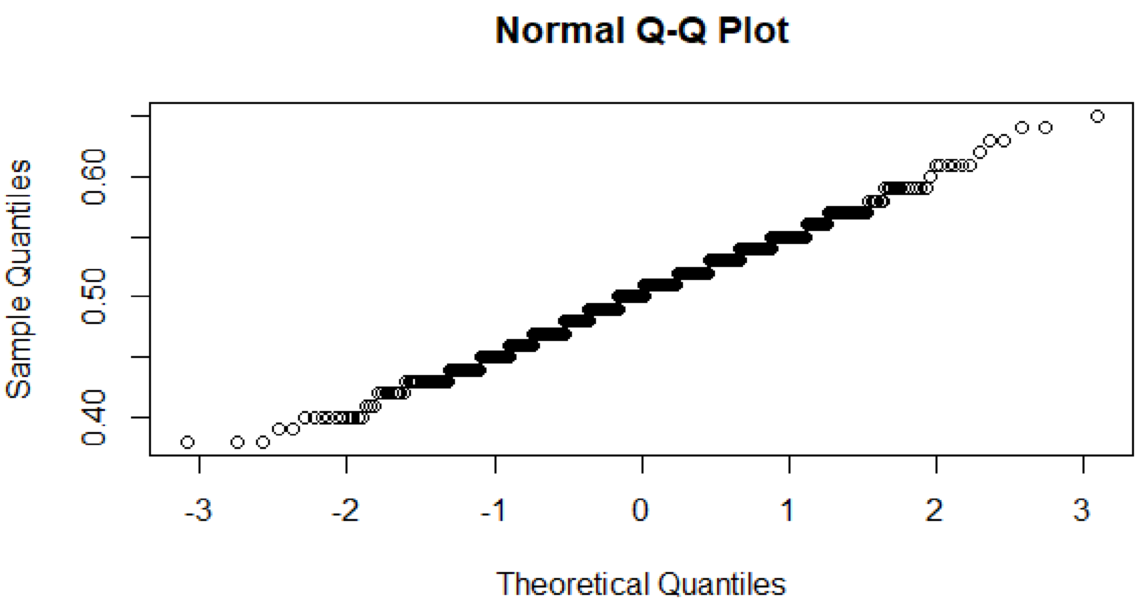
Q2. Suppose X1;X2; : : : ;Xn denotes a random sample from a Bernoulli (p) population, represented by the
random variable X, and let X denote the sample mean. This sample mean also represents the propor-
tion of 1s in the sample, say, ^p. We know from Central Limit Theorem that ^p approximately follows a
normal distribution when n is large. The goal of this exercise is investigate how large n should be for
the approximation to be good. For this investigation, we will focus on p = 0:10; 0:25; 0:50; 0:75; 0:90,
and n = 10; 30; 50; 100.

(a) What is the approximate distribution of ^p when n is large?

Answer:
Approximate distribution of ^p when n is large is Normal Distribution.

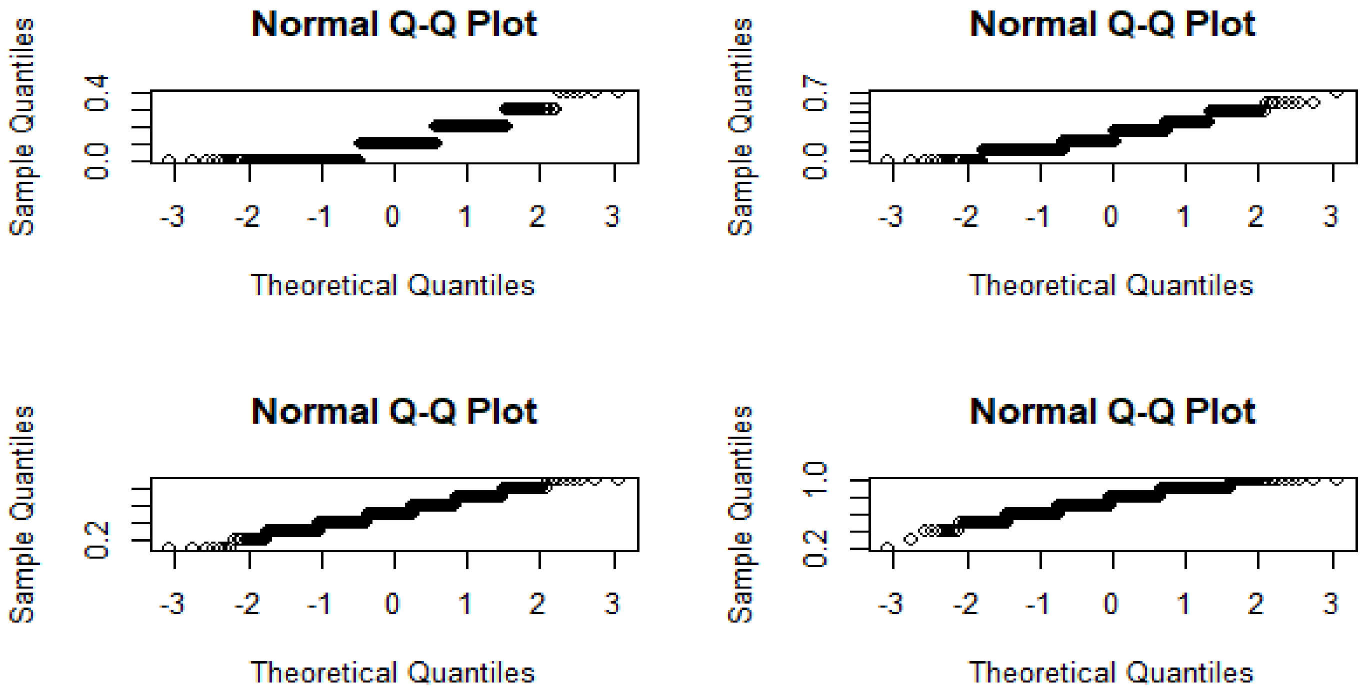
(b) For a given (n; p) combination, simulate 500 values of ^p, and make a normal Q - Q plot of the values. Does the distribution look approximately normal?

Answer:
Taking values n=100, p=0.5
The distribution does look approximately normal.
Code to obtain the plot is provided in Section 2

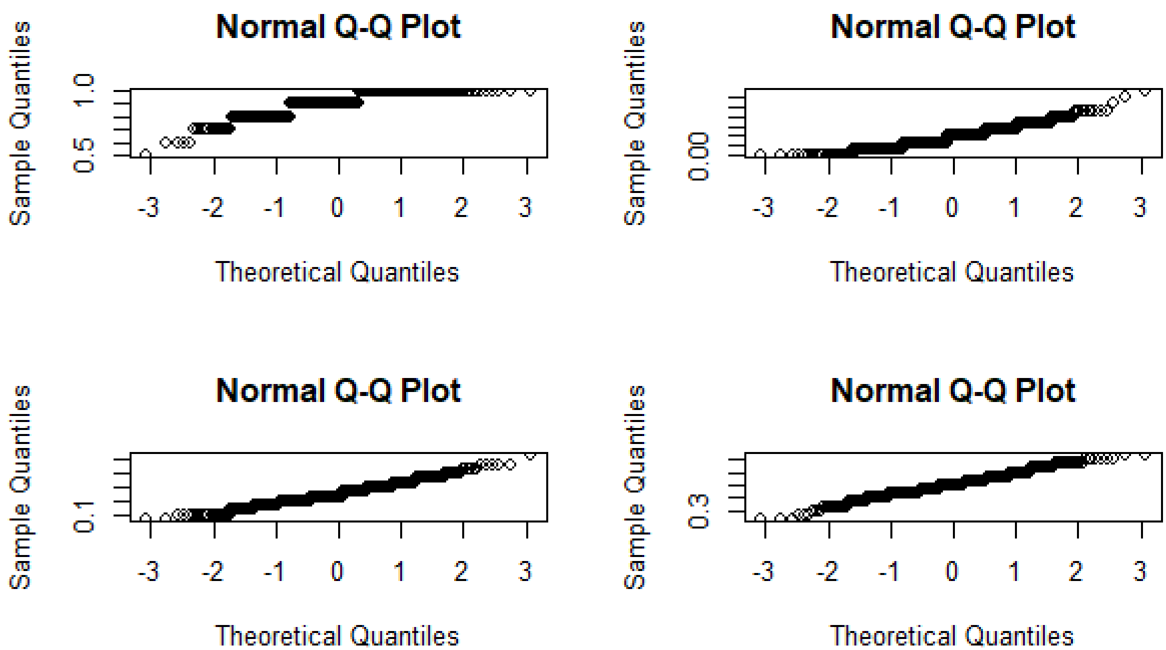


(c) Repeat (b) for the remaining combinations of (n; p) values.

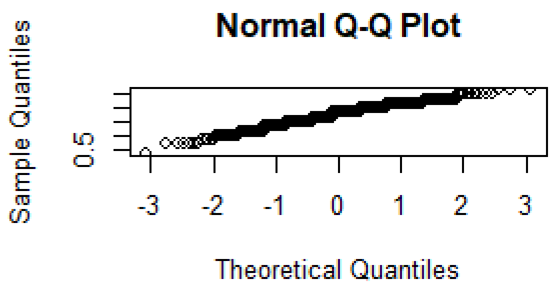
1)n=10, p=0.10
2)n=10, p=0.25
3)n=10, p=0.5
4)n=10, p=0.75

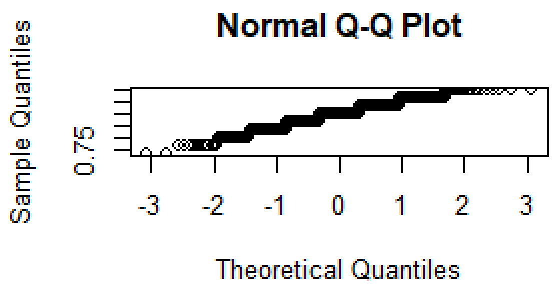


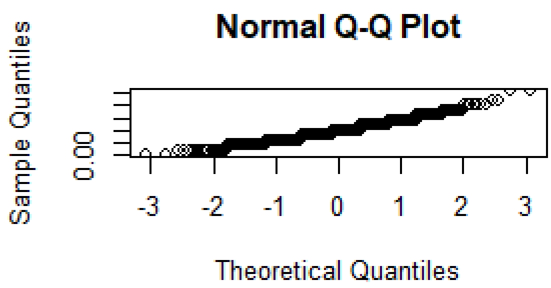
5)n=10, p=0.96)n=30, p=0.10
7)n=30, p=0.25
8)n=30, p=0.5

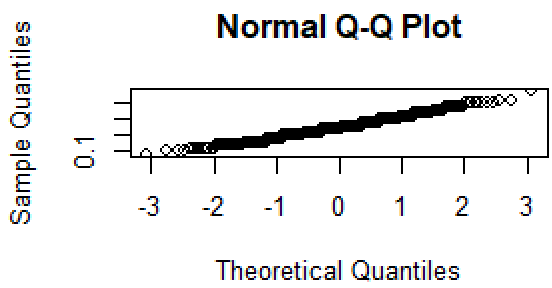


9)n=30, p=0.75
10)n=30, p=0.90
11)n=50, p=0.10
12)n=50, p=0.25

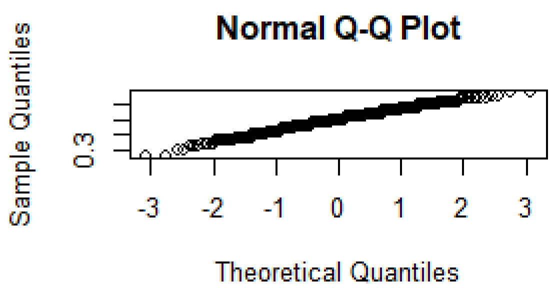


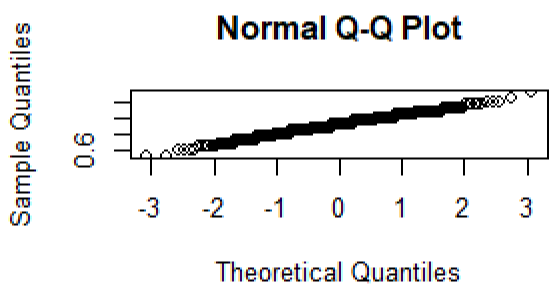


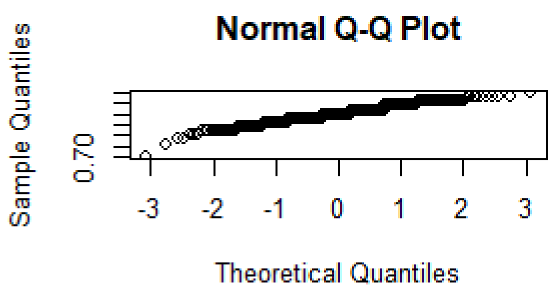


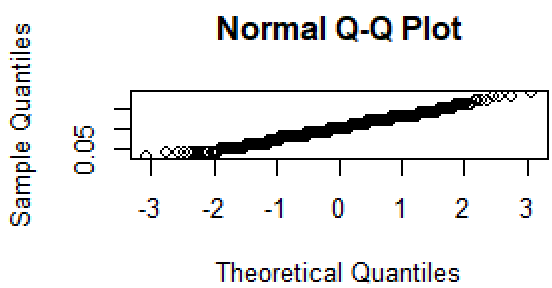


13)n=50, p=0.5
14)n=50, p=0.75
15)n=50, p=0.90
16)n=100, p=0.10

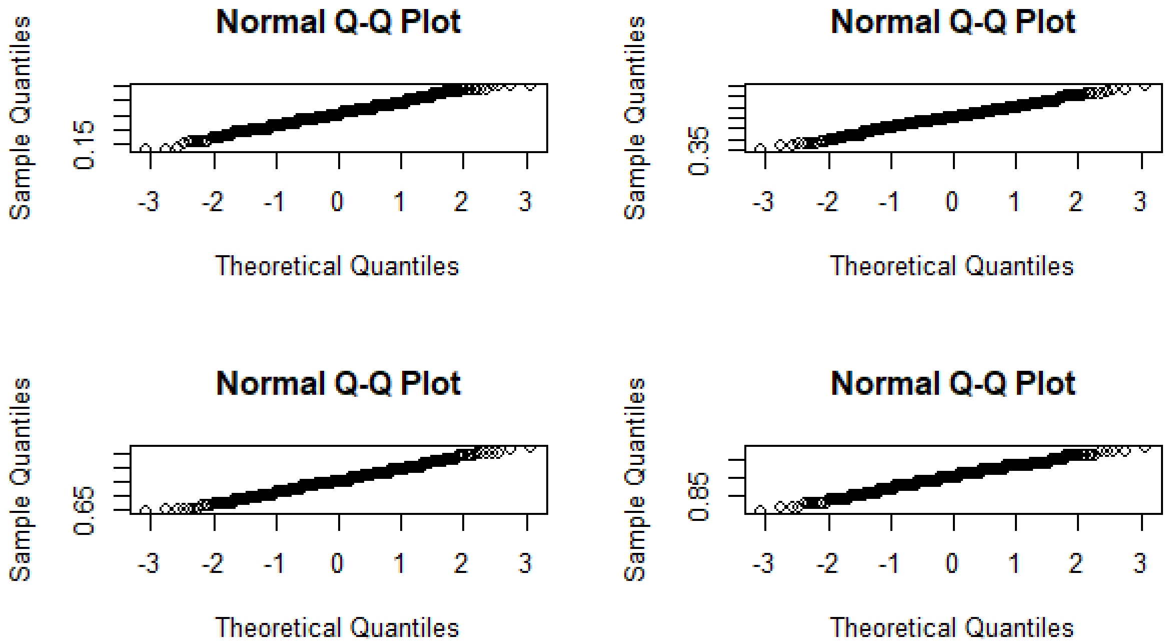








17)n=100, p=0.25
18)n=100, p=0.5
19)n=100, p=0.75
20)n=100, p=0.90



(d) What would you say about how large n should be for the approximation to be good? Does this answer depend on p? Justify your conclusions.

<u>Answer:</u>
According to the results from the above parts b and c we can see that for n>=30 the approximation is good approximation of normal distribution as it almost forms a straight line. In n=10 when probability value is small we get a discrete distribution. The result does depends on p.
1.If the value of p is normal- n value does not have to be too high for good approximation
2. If the value of p is extreme(too high or too low)- n value has to be very high for good approximation

Section 2:

Q1c)MC estimation of E(X), Var(X),P(X<=2) for sample size 1000

```
#possible values if X
x<-1:4

#sample size n
n<-1000

#taking a sample of the given distribution
a=sample(x,n,replace=TRUE)

#calculating mean of distribution
m=mean(a)

#calculating sd of distribution
sd=sqrt(var(a))

#calculating p(x<=2) of distribution
a<-rnorm(a,m,sd)
y<-sum(a[a<=2])
prob=y/n

#replicating final result 1000 times
z=replicate(5,c(mean(a),var(a),prob))
z
```

1d)MC estimation of E(X), Var(X),P(X<=2) for sample size 5000

```
#possible values if X
x<-1:4

#sample size n
n<-5000

#taking a sample of the given distribution
a1=sample(x,n,replace=TRUE)
```

```
#calculating mean of distribution
m1=mean(a1)

#calculating sd of distribution
sd1=sqrt(var(a1))

#calculating p(x<=2) of distribution
a1<-rnorm(a1,m1,sd1)
y1<-sum(a1[a1<=2])
prob1=y1/n

#replicating final result 5000 times
p=replicate(5,c(mean(a1),var(a1),prob))
p
```

1d)MC estimation of E(X), Var(X),P(X<=2) for sample size 10000

```
#possible values if X
x<-1:4

#sample size n
n<-10000

#taking a sample of the given distribution
a2=sample(x,n,replace=TRUE)

#calculating mean of distribution
m2=mean(a2)

#calculating sd of distribution
sd2=sqrt(var(a2))

#calculating p(x<=2) of distribution
a2<-rnorm(a2,m2,sd2)
y2<-sum(a2[a2<=2])
prob=y2/n

#replicating final result 10000 times
q=replicate(5,c(mean(a2),var(a2),prob))
q
```

Q2b)finding qqplot of a (n,p) pair

```
# n=100 p=0.5

#take a random sample, take it's mean. replicate this 500 times

p1 <- replicate(500, mean(x=rbinom(100, 1, 0.5)))
qqnorm(p1)
```

c) The code for part c is the same as part b with different values of n and p

```
#to display four plots in one page
par(mfrow=c(2,2))

#n=10, p=0.1
p1 <- replicate(500, mean(rbinom(10, 1, 0.1)))
qqnorm(p1)



#n=10, p=0.25
p2 <- replicate(500, mean(rbinom(10, 1, 0.25)))
qqnorm(p2)

#n=10, p=0.5
p3 <- replicate(500, mean(rbinom(10, 1, 0.5)))
qqnorm(p3)

#n=10, p=0.75
p4 <- replicate(500, mean(rbinom(10, 1, 0.75)))
qqnorm(p4)

#n=10, p=0.9
p5 <- replicate(500, mean(rbinom(10, 1, 0.9)))
qqnorm(p5)

#n=30, p=0.1
p6 <- replicate(500, mean(rbinom(30, 1, 0.1)))
qqnorm(p6)

#n=30, p=0.25
p7 <- replicate(500, mean(rbinom(30, 1, 0.25)))
qqnorm(p7)

#n=30, p=0.5
```

```
p8 <- replicate(500, mean(rbinom(30, 1, 0.5)))
qqnorm(p8)

#n=30, p=0.75
p9 <- replicate(500, mean(rbinom(30, 1, 0.75)))
qqnorm(p9)

#n=30, p=0.9
p10 <- replicate(500, mean(rbinom(30, 1, 0.9)))
qqnorm(p10)

#n=50, p=0.1
p11 <- replicate(500, mean(rbinom(50, 1, 0.1)))
qqnorm(p11)

#n=50, p=0.25
p12 <- replicate(500, mean(rbinom(50, 1, 0.25)))
qqnorm(p12)


#n=50, p=0.5
p13 <- replicate(500, mean(rbinom(50, 1, 0.5)))
qqnorm(p13)

#n=50, p=0.75
p14 <- replicate(500, mean(rbinom(50, 1, 0.75)))
qqnorm(p14)

#n=50, p=0.9
p15 <- replicate(500, mean(rbinom(50, 1, 0.9)))
qqnorm(p15)

#n=100, p=0.1
p16 <- replicate(500, mean(rbinom(100, 1, 0.1)))
qqnorm(p16)

#n=100, p=0.25
p17 <- replicate(500, mean(rbinom(100, 1, 0.25)))
qqnorm(p17)

#n=100, p=0.5
p18 <- replicate(500, mean(rbinom(100, 1, 0.5)))
qqnorm(p18)

#n=100, p=0.75
```

```
p19 <- replicate(500, mean(rbinom(100, 1, 0.75)))
qqnorm(p19)

#n=100, p=0.9
p20 <- replicate(500, mean(rbinom(100, 1, 0.9)))
qqnorm(p20)
```