Mini Project: 1

Names of Group members: Sashidhar Donthiri and SathyaPooja Ramireddy

Contributions:

1. xi={1,2,3,4}. P(x)={.5, .125, .125, .25}

E(X)= ∑x\*P(x) = 1\*.5+ 2\*.125+ 3\*.125 + 4\*.25 = 2.125

Var(X)=E(x2)-(E(x))2

x2={1,4,9,16}

E(x2)=.5+.5+9/8+4

=6.125

(E(x))2= (17/8)2

Var(X)=1.6093.

1. Here X~f(x), consider it be an arbitrary distribution function, where ∑Pi=1 and P(X=xi)=pi

Assume the value of N be not very large, then we may not consider LLN and CLT.

Lets divide the subinterval between [0,1]

x1=[0,1/2]

x2=[1/2,5/8]

x3=[5/8,6/8]

x4=[6/8,1]

If xi falls in the respective ranges(sub intervals ), then the particular P[X=xi]is assigned.

P[X=xi]=P[U falls in subinterval i]

=P[P0+P1+P2+…Pi-1 < U < P0+P1+P2+P3+…..Pi]

=F(P0+P1+P2….+Pi) – F(P0+P1+….+PI-1)

=P0+P1+P2….+ Pi - P0+P1+….+PI-1

=Pi

Lets assume that the value of N is large, i.e.; large number of draws then,

Simulate a large number of N independent draws from distribution of X1,X2,….,XN

(E) Monte Carlo estimator of µ = (X`)sample average = (1/N)\*(∑Xi)

(Variance) Mote Carlo estimator of σ2= E((X- µ)2) = (1/(N-1))\* (∑Xi- x`)2

1. A=replicate(1000,sample(c(1,2,3,4),1,TRUE,c(.5,.125,.125,.25)))

|  |  |  |  |
| --- | --- | --- | --- |
| n | Mean | Variance | P(X<=2) (cdf) (∑P(X=xi))/N |
| 1000 | 2.144 | 1.668933 | .003 |
| 1000 | 2.168 | 1.665441 | .003 |
| 1000 | 2.066 | 1.581225 | .003 |
| 1000 | 2.074 | 1.544068 | .003 |
| 1000 | 2.144 | 1.604869 | .003 |

|  |  |  |  |
| --- | --- | --- | --- |
| N | Mean | Variance | P(X<=2) (cdf) |
| 5000 | 2.1396 | 1.601232 | 0.00625 |
| 5000 | 2.109 | 1.590237 | 0.00625 |
| 5000 | 2.092 | 1.590417 | 0.00625 |
| 5000 | 2.428 | 1.619932 | 0.00625 |
| 5000 | 2.1576 | 1.613885 | 0.00625 |

|  |  |  |  |
| --- | --- | --- | --- |
| N | Mean | Variance | P(X<=2) (cdf) |
| 10000 | 2.1278 | 1.610628 | .0003 |
| 10000 | 2.1272 | 1.612381 | .0003 |
| 10000 | 2.1505 | 1.627412 | .0003 |
| 10000 | 2,1091 | 1.601957 | .0003 |
| 10000 | 2.1299 | 1.599586 | .0003 |

1. From the results obtained by a,c,d we observe that as the value of N increases the distribution turns out to be normal and the values of µ and σ2 are almost equal to the true values of µ and σ2.

Also the variance decreases as value of N increases , thus follows LLN.

2)

a) For large values of n , ̂ **p**-**hat** follows a normal distribution.

Explanation: By **CLT**, for large n, **p**-**hat** ~= N[E(**p**-**hat** )=p , var(**p**-**hat** ) = p(1-p)/n]

Or Z=( **p**-**hat**)-(p)/sqrt(p(1-p)/n) ~= N(0,1).

b) For a given (n, p) combination, simulate 500 values of ˆp, and make a normal Q − Q plot of the values, the

R-Code:

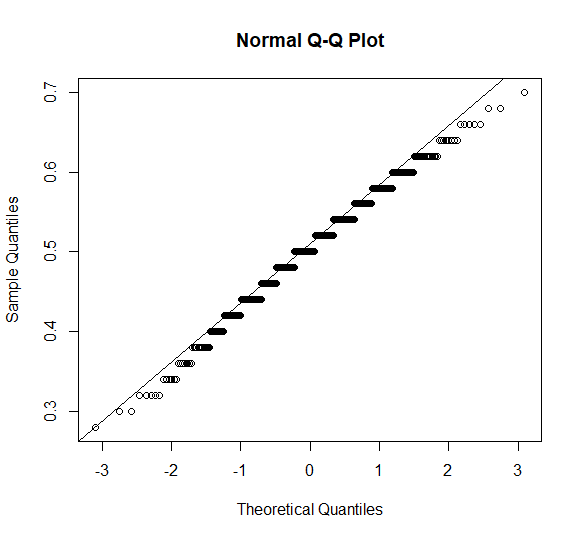
#replicate it for 500 times, following a binomial distribution

p.10k= replicate(500,mean(rbinom(10,1,.1)))

qqnorm(p.10k)

qqline(p.10k)

From the graph obtained, it is clear that for a given (n,p) the given distribution does not look like normal distribution, because the value of N is comparably small.



c) Plots for the combination of (n,p) values = (10,(.1,.25,.5,.75,.90))

R-Code:

#Code for each of (n,p) combinations

p.10k= replicate(500,mean(rbinom(10,1,.1)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(10,1,.25)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(10,1,.50)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(10,1,.75)))

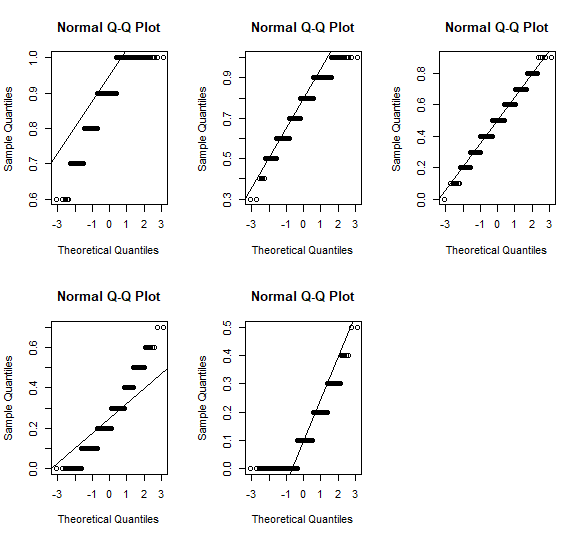
qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(10,1,.90)))

qqnorm(p.10k)

qqline(p.10k)



Plots for the combination of (n,p) values = (30,(.1,.25,.5,.75,.90)), (50,(.1,.25,.5,.75,.9))

R-Code:

#Code for each of (n,p) combinations

p.10k= replicate(500,mean(rbinom(30,1,.1)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(30,1,.25)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(30,1,.50)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(30,1,.75)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(30,1,.90)))

qqnorm(p.10k)

qqline(p.10k)

R-Code:

#Code for each of (n,p) combinations

p.10k= replicate(500,mean(rbinom(50,1,.1)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(50,1,.25)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(50,1,.50)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(50,1,.75)))

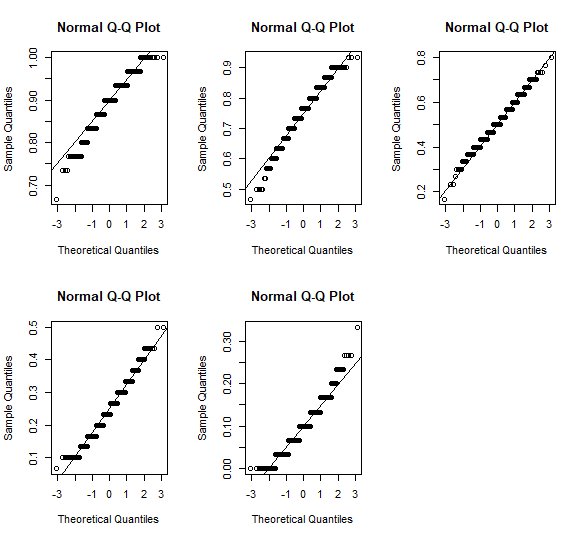
qqnorm(p.10k)

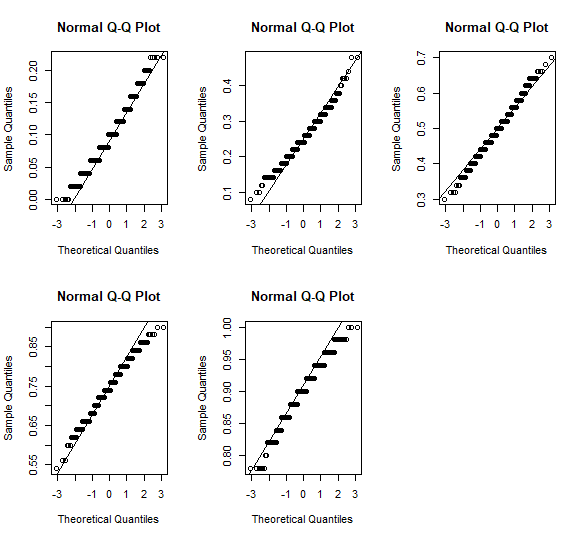
qqline(p.10k)

p.10k= replicate(500,mean(rbinom(50,1,.90)))

qqnorm(p.10k)

qqline(p.10k)





Plots for the combination of (n,p) values = (30,(.1,.25,.5,.75,.90)), (50,(.1,.25,.5,.75,.9))

R-Code:

#Code for each of (n,p) combinations

p.10k= replicate(500,mean(rbinom(100,1,.1)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(100,1,.25)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(100,1,.50)))

qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(100,1,.75)))

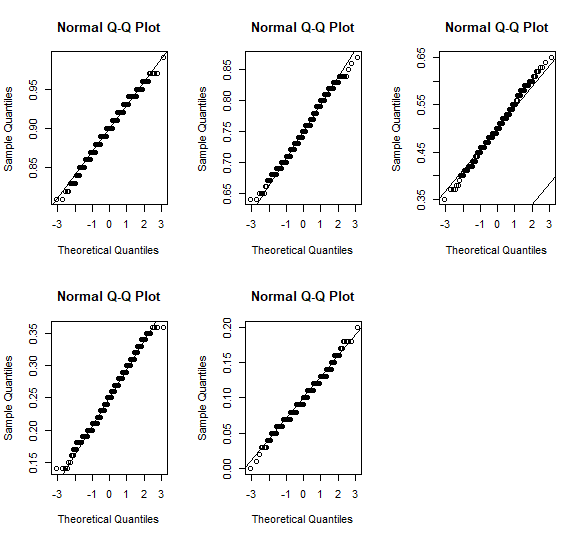
qqnorm(p.10k)

qqline(p.10k)

p.10k= replicate(500,mean(rbinom(100,1,.90)))

qqnorm(p.10k)

qqline(p.10k)



d) From the observations in (c), it is evident that the distribution is normal as the value of N increases following LLN. The value of N can be calculated by the calculating N~= Z2α/2 \*p(1-p)/epsilon2

Here, the conclusion depends on the value of p. So we have to replace p value with a good guess say p\*.

So, to get the max value of N we have to replace p(1-p) with max value ~1/4.

* N~= Z2α/2 \*1/4/Epsilon2
* Suppose desired accuracy is, (Epsilon, α)=(0.02,0.05)
* Then N~= 1.96\*2/4/0.032
* N~1068(larger value of N)

From above statements, we can conclude that the value of N should be around 1068 for good approximation also it depends on value of p.

Graph for p.10k=replicate(500,mean(rbinom(1068,1,.25/.1/.75/.5))) shows that as N=1068, the approximation is best fit for the distribution.

