Exercise-1: Simulation of various distributions

I. Uniform Distribution:

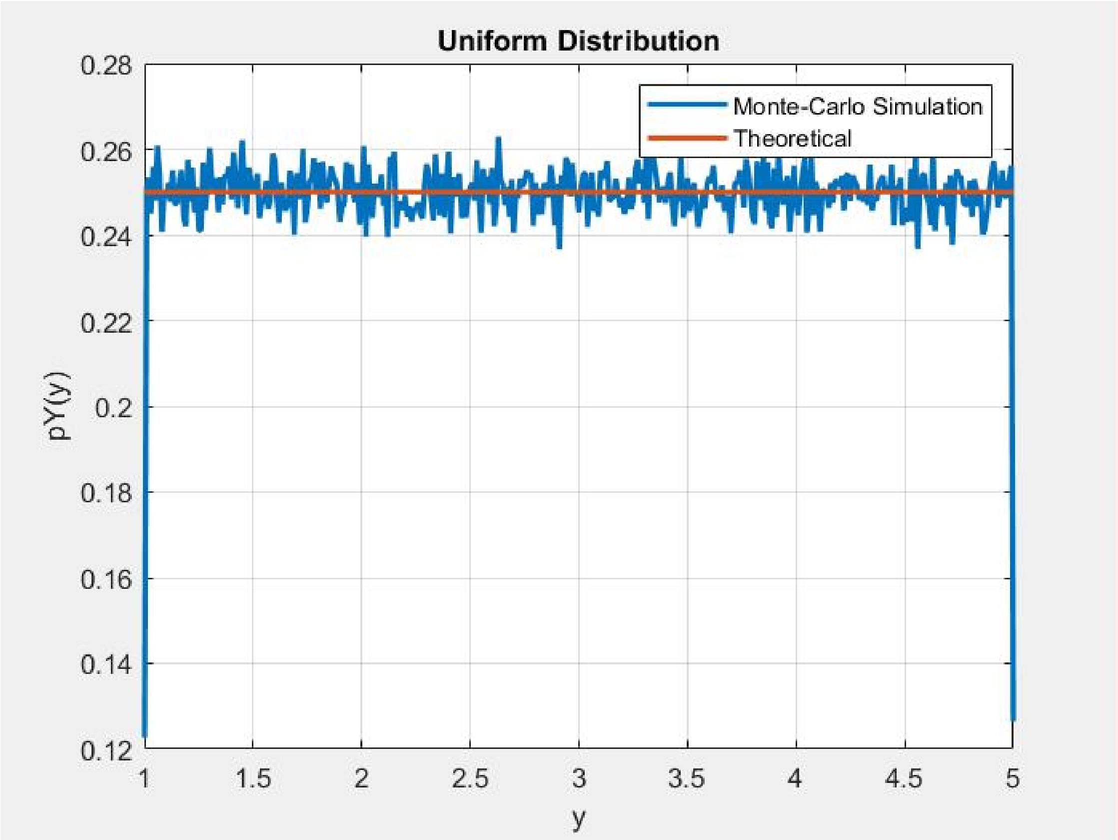
➢ Program:

```
clearvars; close all;
Nexperiments = 1000000;
yMin = 0; yMax = 1;
y = yMin + (yMax-yMin)*rand(1,Nexperiments);
stepsize = 0.01;
yBins = yMin:stepsize:yMax;
yEdges = [yMin-stepsize/2 yBins+stepsize/2];
nY = histc(y,yEdges);
pYsimulated = nY/stepsize/Nexperiments;
pYtheoretical = unifpdf(yBins,yMin,yMax);
figure;
plot(yBins,[pYsimulated(1:end-1); pYtheoretical],'linewidth',2);
xlabel('y'); ylabel('pY(y)'); title('Uniform Distribution');
legend('Monte-Carlo Simulation','Theoretical');
grid;
yMeanSimulated = mean(y);
yVarianceSimulated = var(y);

yMeanTheoretical = (yMin+yMax)/2;
yVarianceTheoretical = (yMax-yMin)^2/12;
fprintf(1,'Mean value of uniformly distributed random variable: %1.2f (simulated), %1.2f
(theoretical)\n',yMeanSimulated,yMeanTheoretical);
fprintf(1,'Variance of uniformly distributed random variable: %1.2f (simulated), %1.2f
(theoretical)\n',yVarianceSimulated,yVarianceTheoretical);
```

➢ Output:

```
Command Window
>> T1
Mean value of uniformly distributed random variable: 0.50 (simulated), 0.50 (theoretical)
Variance of uniformly distributed random variable: 0.08 (simulated), 0.08 (theoretical)
>> T1
Mean value of uniformly distributed random variable: 3.00 (simulated), 3.00 (theoretical)
Variance of uniformly distributed random variable: 1.33 (simulated), 1.33 (theoretical)
```



➢ Inference:

The Uniform Probability Density Function

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \le x \le b \\ 0 & \text{otherwise} \end{cases}$$

f(x) = value of density function at any x value
a = minimum value of x
b = maximum value of x
It is clear that for uniform distribution the simulated and theoretical PDF is same as we increase Nexperiments.

II. Gaussian Distribution:

➢ Program:

```
clearvars; close all;
muGauss = 0; varGauss = 1; stdGauss = sqrt(varGauss);
Nexperiments = 100000;
```

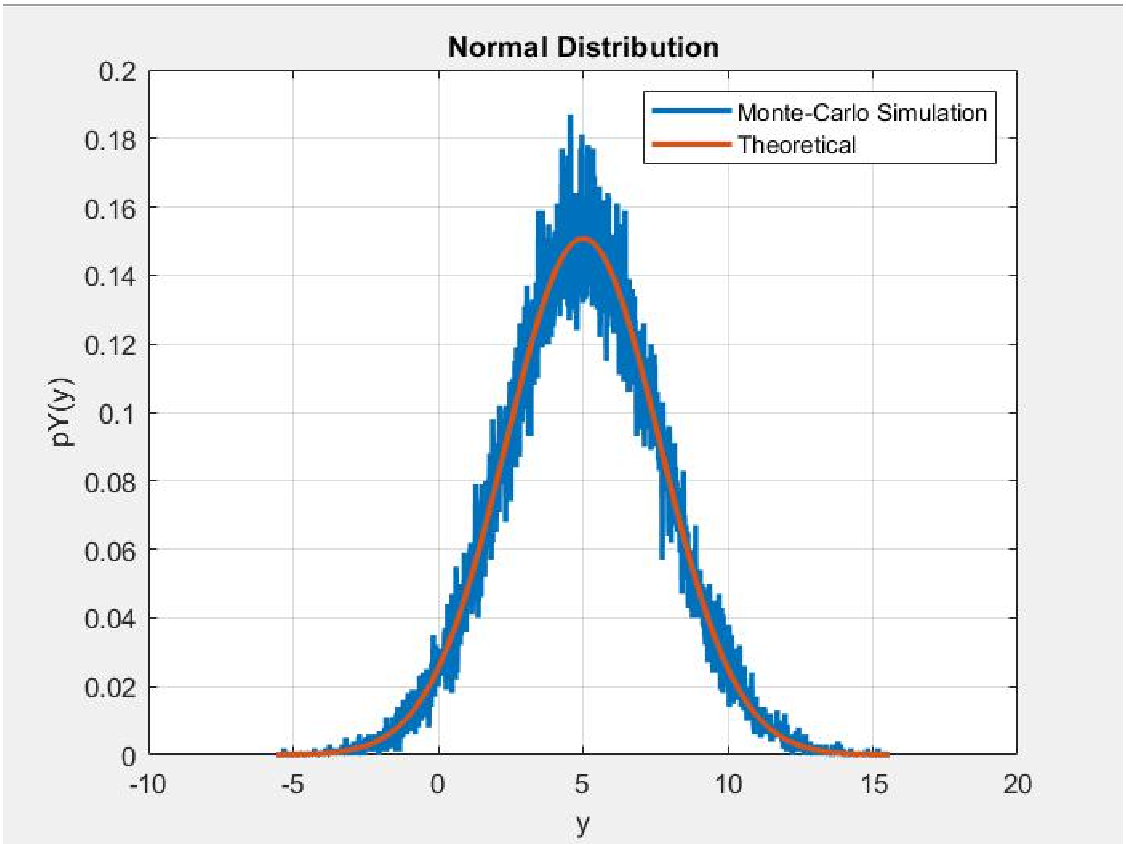
```
yMin = muGauss-4*stdGauss; yMax = muGauss+4*stdGauss;
y = stdGauss*randn(1,Nexperiments)+ muGauss;
stepsize = 0.01;
yBins = yMin:stepsize:yMax;
yEdges = [yMin-stepsize/2 yBins+stepsize/2];
nY = histc(y,yEdges);
pYsimulated = nY/stepsize/Nexperiments;
pYtheoretical = normpdf(yBins,muGauss,stdGauss);
figure;
plot(yBins,[pYsimulated(1:end-1); pYtheoretical],'linewidth',2);
xlabel('y'); ylabel('pY(y)'); title('Normal Distribution');
legend('Monte-Carlo Simulation','Theoretical');
grid;
yMeanSimulated = mean(y);
yVarianceSimulated = var(y);

yMeanTheoretical = 0;
yVarianceTheoretical = 1;
fprintf(1,'Mean value of uniformly distributed random variable: %1.2f (simulated), %1.2f
(theoretical)\n'...
,yMeanSimulated,yMeanTheoretical);
fprintf(1,'Variance of uniformly distributed random variable: %1.2f (simulated), %1.2f
(theoretical)\n',...
yVarianceSimulated,yVarianceTheoretical);
```

- **Output:**

Command Window

```
>> T1
Mean value of uniformly distributed random variable: 4.98 (simulated), 0.00 (theoretical)
Variance of uniformly distributed random variable: 7.06 (simulated), 1.00 (theoretical)
>> T1
Mean value of uniformly distributed random variable: 5.01 (simulated), 0.00 (theoretical)
Variance of uniformly distributed random variable: 7.03 (simulated), 1.00 (theoretical)
```



➢ Inference:

The continuous Gaussian Distribution

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where σ is the standard deviation and μ the mean

It is clear that for Gaussian Distribution the simulated and theoretical PDF tends to be same when Nexperiments tend to infinity.

III. Binomial Distribution:

➢ Program:

```
clearvars; close all;
Nexperiments = 100000;
N = 20;
```

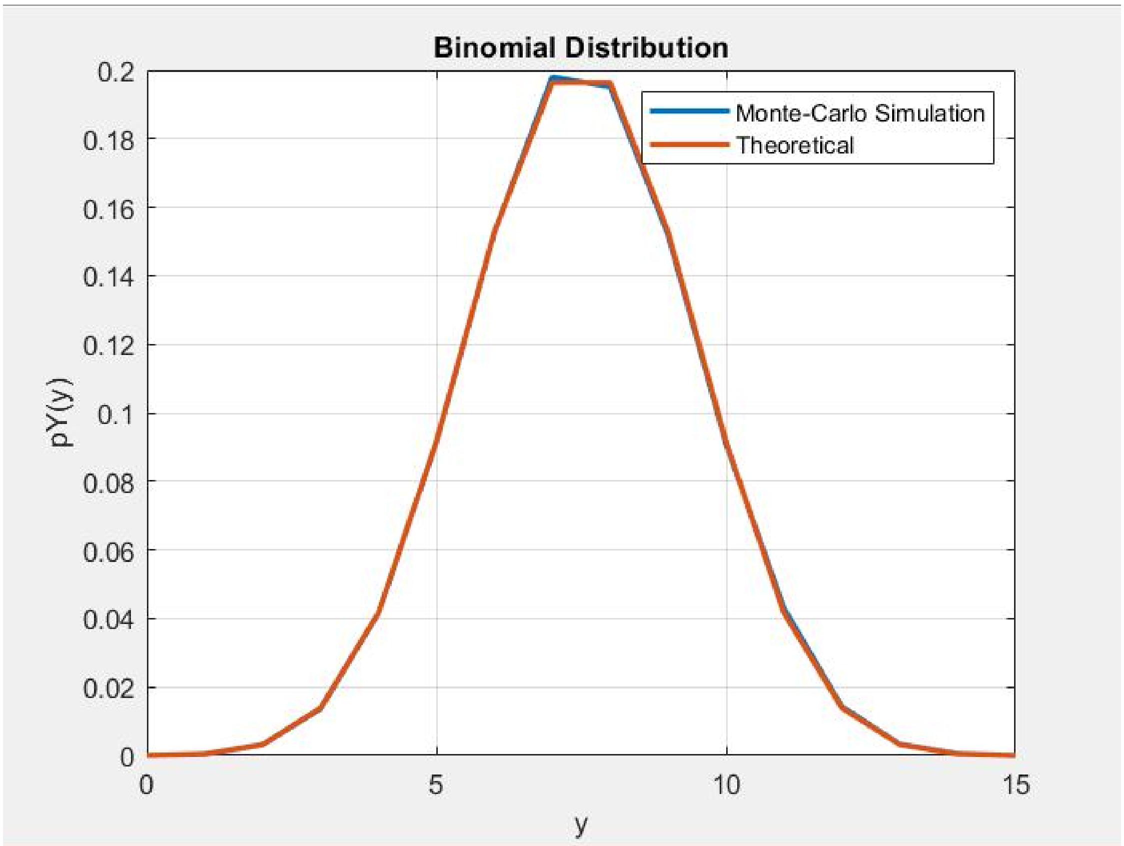
```
yBins = 0:N;
yEdges = [-0.5 yBins+0.5];
p = 0.5;
y = zeros(1,Nexperiments);
for kk = 1:Nexperiments
b = double(rand(1,N)>(1-p));
y(kk) = sum(b);
end
nY = histc(y,yEdges);
pYsimulated = nY/Nexperiments;
pYtheoretical = binopdf(yBins,N,p);
figure;
plot(yBins,[pYsimulated(1:end-1); pYtheoretical],'linewidth',2);
xlabel('y'); ylabel('pY(y)'); title('Binomial Distribution');
legend('Monte-Carlo Simulation','Theoretical');
grid;
yMeanSimulated = mean(y);
yVarianceSimulated = var(y);

yMeanTheoretical = N*p;
yVarianceTheoretical = N*p*(1-p);
fprintf(1,'Mean value of Binomially distributed random variable: %1.2f (simulated), %1.2f
(theoretical)\n',yMeanSimulated,yMeanTheoretical);

fprintf(1,'Variance of Binomially distributed random variable: %1.2f (simulated), %1.2f
(theoretical)\n',yVarianceSimulated,yVarianceTheoretical);
```

- **Output:**

```
Command Window
>> T1
Mean value of Binomially distributed random variable: 7.50 (simulated), 7.50 (theoretical)
Variance of Binomially distributed random variable: 3.76 (simulated), 3.75 (theoretical)
>> T1
Mean value of Binomially distributed random variable: 7.51 (simulated), 7.50 (theoretical)
Variance of Binomially distributed random variable: 3.76 (simulated), 3.75 (theoretical)
```



➢ Inference:

The Binomial Distribution formula

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

f(x) = probability of x successes in n trials, with probability of success p on each trial

x = no. of successes in sample

n = sample size

It is clear that for Gaussian Distribution the simulated and theoretical PDF is same as increase in no. of experiments.

Exercise-2:

1. Uniform Distribution:

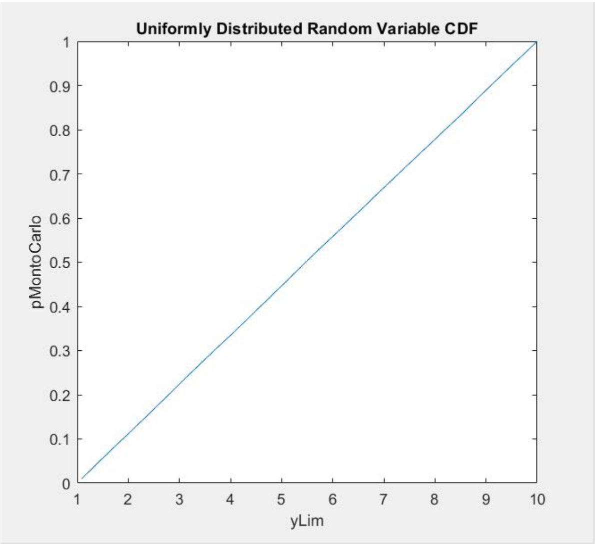
➢ Program:

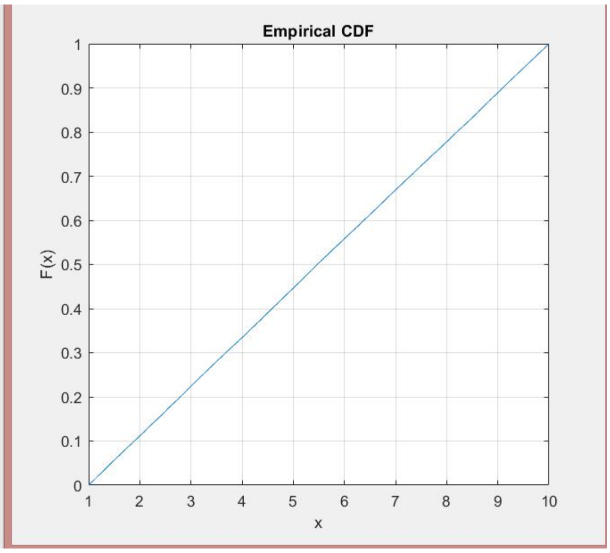
```
clearvars; close all;
Nexperiments = 100000;

yMin = 1; yMax = 10; %change these values and observe the effect
y = yMin + (yMax-yMin)*rand(1,Nexperiments);
yLim = yMin+(1/100)*(yMax-yMin);
pMonteCarlo= sum(y<yLim)/Nexperiments;
for i=2:100
yLim = [yLim ;yMin+(i/100)*(yMax-yMin)];
pMonteCarlo= [pMonteCarlo; sum(y<yLim(i))/Nexperiments];
end
plot(yLim,pMonteCarlo);
xlabel('yLim'); ylabel('pMontoCarlo');
title('Uniformly Distributed Random Variable CDF')

% %Theoretical%
figure;
cdfplot(y);
```

➢ Output:





➢ Inference:

The cumulative distribution function of a uniform random variable x is:

$$F(x) = \begin{cases} 0 & \text{for } x < a \\ \frac{x-a}{b-a} & \text{for } a \le x \le b \\ 1 & \text{for } x > b \end{cases}$$

It is clear that for Uniform Distribution the simulated and theoretical CDF is same.

2. Gaussian Distribution:

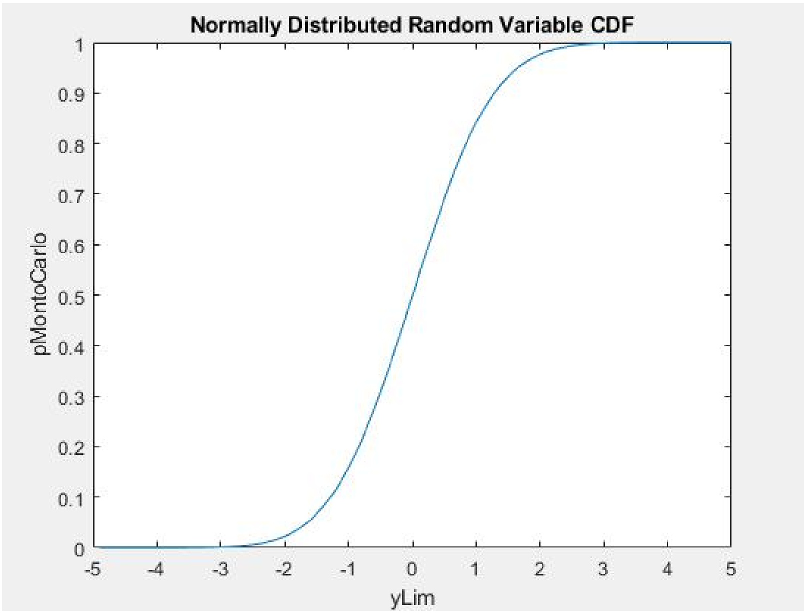
➢ Program:

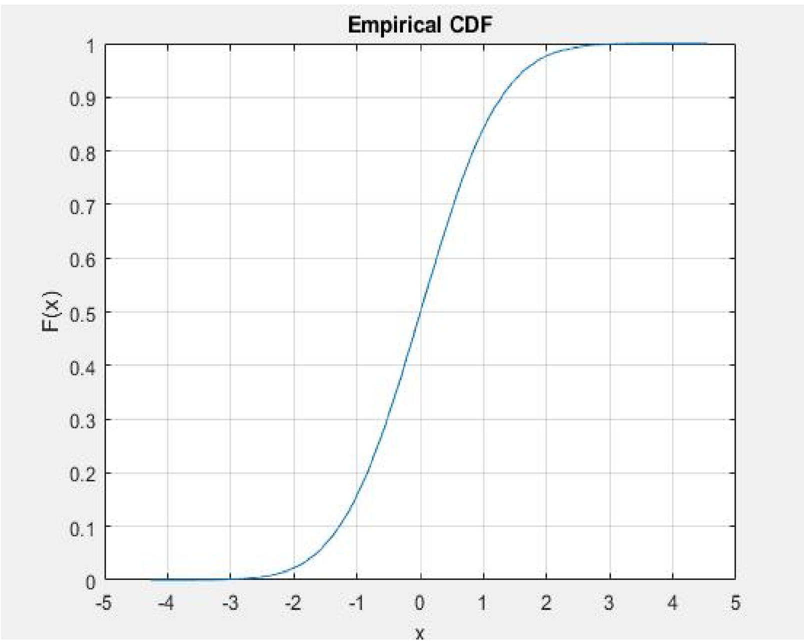
```
clearvars; close all;
Nexperiments = 100000;
muGauss = 0; varGauss = 1; stdGauss = sqrt(varGauss);
yMin = muGauss-5*stdGauss; yMax = muGauss+5*stdGauss;
y = stdGauss*randn(1,Nexperiments)+ muGauss;
yLim = yMin+(1/100)*(yMax-yMin);
pMonteCarlo= sum(y<yLim)/Nexperiments;
for i=2:100
yLim = [yLim ;yMin+(i/100)*(yMax-yMin)];
pMonteCarlo= [pMonteCarlo; sum(y<yLim(i))/Nexperiments];
end
plot(yLim,pMonteCarlo);
xlabel('yLim'); ylabel('pMontoCarlo');
title('Normally Distributed Random Variable CDF');


%Simulated%


figure;
cdfplot(y);
```

- ## ***Output:***


Normally Distributed Random Variable CDF
pMontoCarlo
yLim
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-5
-4
-3
-2
-1
0
1
2
3
4
5


Empirical CDF
F(x)
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-5
-4
-3
-2
-1
0
1
2
3
4
5

➢ Inference:

$$\text{Cumulative Distribution Function (cdf)} = F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} dx$$

Where μ is the population mean and σ is the population standard deviation;

It is clear that for Gaussian Distribution the simulated and theoretical CDF is same.

3. Binomial Distribution

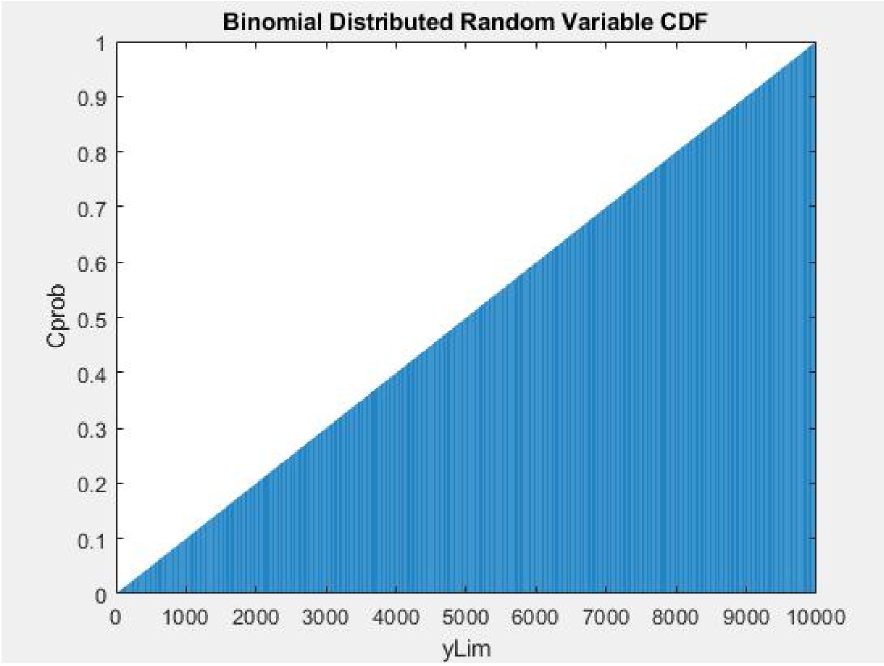
➢ Program:

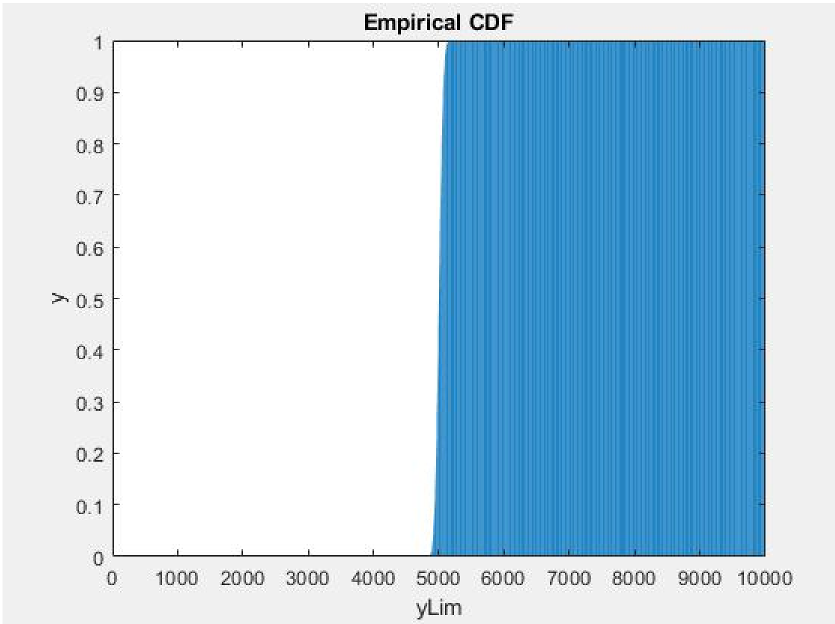
```
clearvars; close all;
Nexperiments = 1000;
% N = 10;
p = 0.5;
b = binornd(1,p,1,Nexperiments);
yLim=1;
for r=2:Nexperiments
    yLim=[yLim;r];
end
Cprob= 1/sum(b);

for i=2:Nexperiments
  Cprob= [Cprob;(i*(1/Nexperiments))];
end
 bar(Cprob);
 xlabel('yLim'); ylabel('Cprob');
 title('Binomial Distributed Random Variable CDF');
% plot(yLim,Cprob);
%%
%Theoretical

y = binocdf(yLim,Nexperiments,p);
figure;
bar(y);
xlabel('yLim'); ylabel('y');
title('Empirical CDF');
```

➢ Output:





- **Inference:**

The cumulative Distribution function for Binomial Distribution:

$$\sum_{k=0}^{x} \frac{n!}{k!\,(n-k)!} p^k (1-p)^{n-k}$$

It is clear that for Binomial Distribution the simulated and theoretical CDF is nearly same.

Exercise-3: Simulation of Binary Symmetric Channel BSC(p)

1) Using BSC(p) Channel:

➢ Program:

```
clearvars; close all;
N = 10000;
s = binornd(1,0.5,1,10000);
p = input('Enter probability of error: ');

if(p == 0.01)
   n=binornd(1,0.01,1,10000);
elseif(p == 0.1)
   n=binornd(1,0.1,1,10000);
else
   n=binornd(1,0.2,1,10000);
end

r= xor(s,n);
q=0;

for kk = 1:10000

if(r(1, kk)~= s(1, kk))
   q=q+1;
end

end
error= q/10000;
fprintf(1,'Probability of error using BSC(p) channel: %1.4f (simulated), %1.2f
(theoretical)\n'...'
,error,p);
```

➢ Output:

```
>> Lab_3_1_bsc
Enter probability of error: 0.01
Probability of error using BSC(p) channel: 0.0100 (simulated), 0.01 (theoretical)
>> Lab_3_1_bsc
Enter probability of error: 0.1
Probability of error using BSC(p) channel: 0.0977 (simulated), 0.10 (theoretical)
>> Lab_3_1_bsc
Enter probability of error: 0.2
Probability of error using BSC(p) channel: 0.1997 (simulated), 0.20 (theoretical)
```

➢ Inference:

It is clear that for binary symmetric channel BSC(p), both values of simulation and theoretical comes nearly same.

2) Using Majority Vote Decoder:

➢ Program:

```
clearvars; close all;
Nexperiments = 10000;
N = 10000;
s = binornd(1,0.5,10000,1);
for i= 1:10000
   if(s(i,1)==1)
      s(i,2)=1;
      s(i,3)=1;
   end
end
%n=binornd(1,0.1,10000,3);
%p = 0.1;
p = input('Enter probability of error: ');
if(p == 0.01)
   n=binornd(1,0.01,10000,3);
elseif(p == 0.1)
   n=binornd(1,0.1,10000,3);
else
   n=binornd(1,0.2,10000,3);
end
r= xor(s,n);
q=0;
for kk = 1:10000
   z=0;
 for jj=1:3
    if(r(kk,jj)~= s(kk,jj))
       z=z+1;
    end
 end
 if(z>=2)
    q=q+1;
 end
end
error= q/10000;
fprintf(1,'Probability of error using BSC(p) channel, Majority Vote Decoder: %1.4f
(simulated), %1.2f (theoretical)\n'...'
,error,p);
```

➢ Output:

```
>> Lab_3_2_majority
Enter probability of error: 0.01
Probability of error using BSC(p) channel, Majority Vote Decoder: 0.0003 (simulated), 0.01 (theoretical)
>> Lab_3_2_majority
Enter probability of error: 0.1
Probability of error using BSC(p) channel, Majority Vote Decoder: 0.0268 (simulated), 0.10 (theoretical)
>> Lab_3_2_majority
Enter probability of error: 0.2
Probability of error using BSC(p) channel, Majority Vote Decoder: 0.1054 (simulated), 0.20 (theoretical)
```

➢ Inference:

It is clear that for binary symmetric channel BSC(p) Majority Vote Decoder, both values of simulation and theoretical comes nearly same.