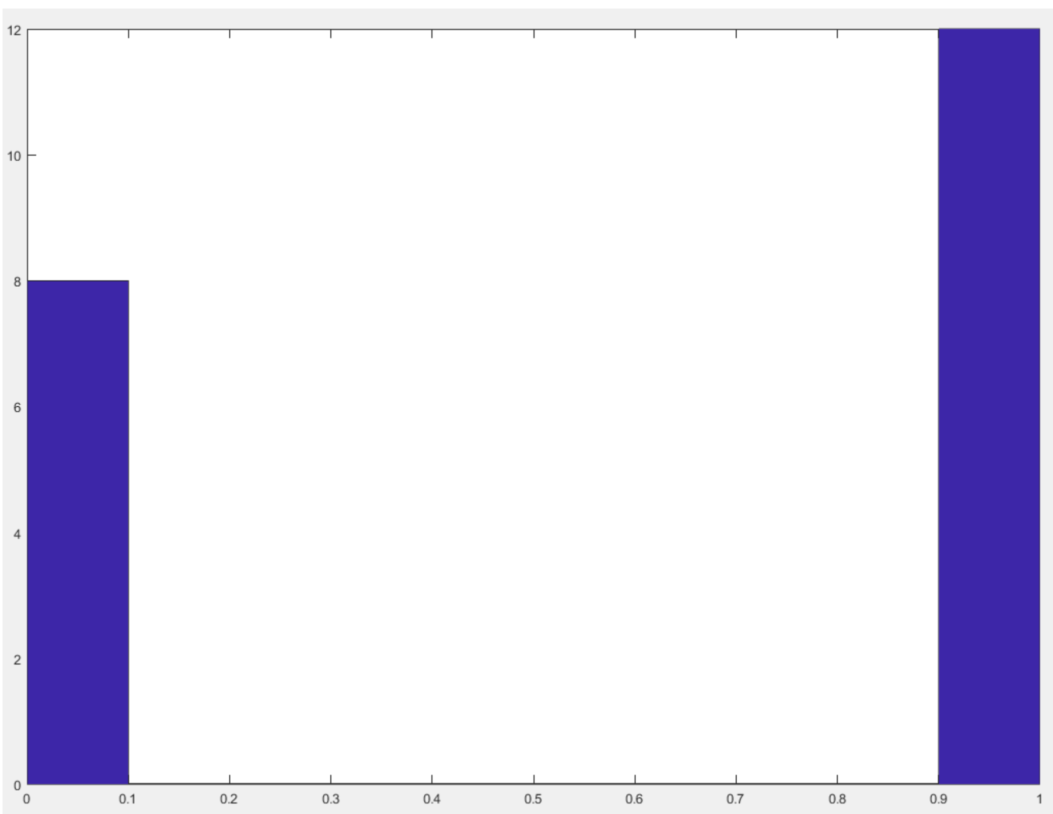
1

```
clc;
clear;
%   1.a
x=binornd(1,0.6,[1,20])
%   1.b
hist(x);
% Not always showing the output exactly to the probability of "p".
% Histogram plot plots number of occurances of the same value.
```

```
x =

  Columns 1 through 18

     1     1     0     0     1     1     1     0     0     0     1     0     0     1     1     1     0     1

  Columns 19 through 20

     1     1
```



my_bernoulli.m | Main.m

```
%   1.c
function x= my_bernoulli(p,N,a,b);

x=rand(1,N);

if or(a,b)==0
   if a>b
        x(x<(1-p))=b;
        x(x>=(1-p))=a;
        x
        hist(x);
   elseif a<b
        x(x<(p))=a;
        x(x>=(p))=b;
        x
        hist(x);
   else
    display('Not a bernoulli Random variable');
   end

else
    if a>b
        x(x>=(1-p))=a;
        x(x<(1-p))=b;
        x
        hist(x);
    elseif a<b
        x(x>=(p))=b;
        x(x<(p))=a;
        x
        hist(x);
    else
        display('Not a bernoulli Random variable');
   end

end
```

my_bernoulli.m | Main.m*

```
clc;
clear;

x=my_bernoulli(0.6,20,1,0);

%       Analytical mean = mx
%       Analytical Variance = Vx

%       2.1.a
%            mx = p = 0.6
%            vx = p(1-p) = 0.6*0.4 = 0.24

mx=0.6;
vx=0.24;

%       System mean = mean_x
%       System Variance = var_x

mean_x = mean(x);
var_x = var(x);

%       Sample Mean = sam_mean_x
%       Sample Variance = sam_var_x

N=length(x)
sam_mean_x = sum(x)/N;
sam_var_x = sum((x-sam_mean_x).^2)/(N-1);



z=[mx vx mean_x var_x   sam_mean_x   sam_var_x]
display('      mx       vx         mean(x)    var(x)   sample mean(x)   sample var(x)')

figure
bar(z)
xlabel('mx  vx  mean(x)  var(x)  sample mean(x)   sample var(x)')
ylabel('Value')

%   The smaple mean and variance are same as the matlab function mean and
%   variance
```

2.1.b

```
N =

    20


z =

    0.6000    0.2400    0.5000    0.2632    0.5000    0.2632

      mx       vx         mean(x)    var(x)   sample mean(x)   sample var(x)
```

Yes, Sample mean and variance and Mat lab functions mean and variance are same.

2.1.c

```
N =

   100


z =

    0.6000    0.2400    0.5600    0.2489    0.5600    0.2489

      mx        vx         mean(x)    var(x)   sample mean(x)   sample var(x)
```

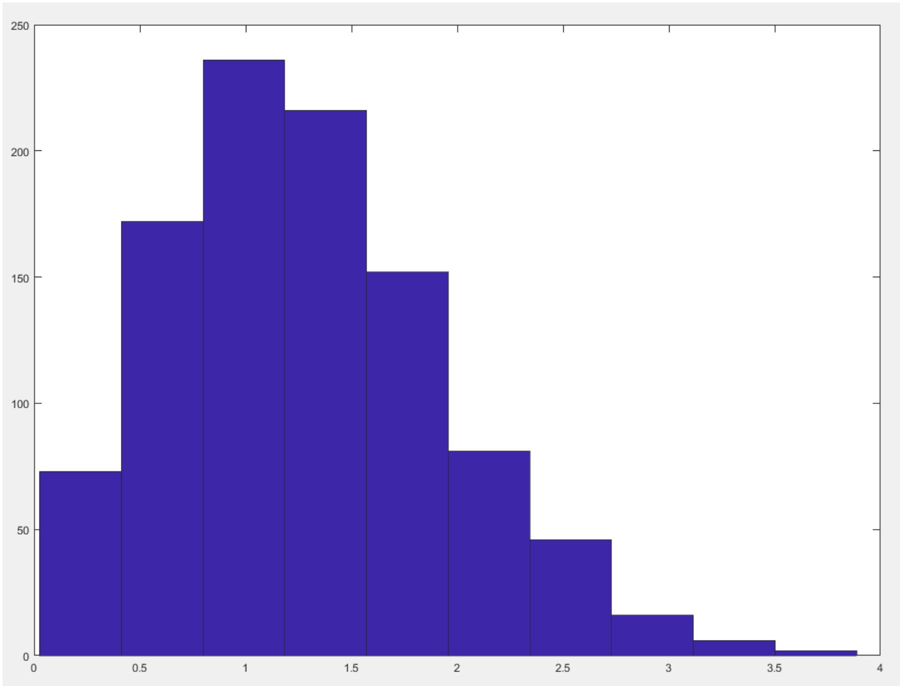
```
N =

        1000


z =

    0.6000    0.2400    0.5960    0.2410    0.5960    0.2410

      mx        vx         mean(x)    var(x)   sample mean(x)   sample var(x)
fx >>
```

Yes, sample mean and variance changes with the value of N.

Yes, they are closer to the analytical values. More the N, closer to the analytical values.

```
%%   3.1.a
%       Mean = sig*sqrt(pi/2)  = 1.253*sig
%       Var =  ((4-pi)/2)*sig^2 = 0.429*sig^2
%%   3.1.b
%       sig^2 = 1

sig = 1;
N=1000;
U=rand(1,N);

x = sqrt(-2*(sig^2)*log(1-U));

samp_mean_x = sum(x)/N
sam_var_x = sum((x-samp_mean_x).^2)/(N-1)

hist(x);
```

```
samp_mean_x =

    1.2840


sam_var_x =

    0.4246
```



Rayleigh.m*

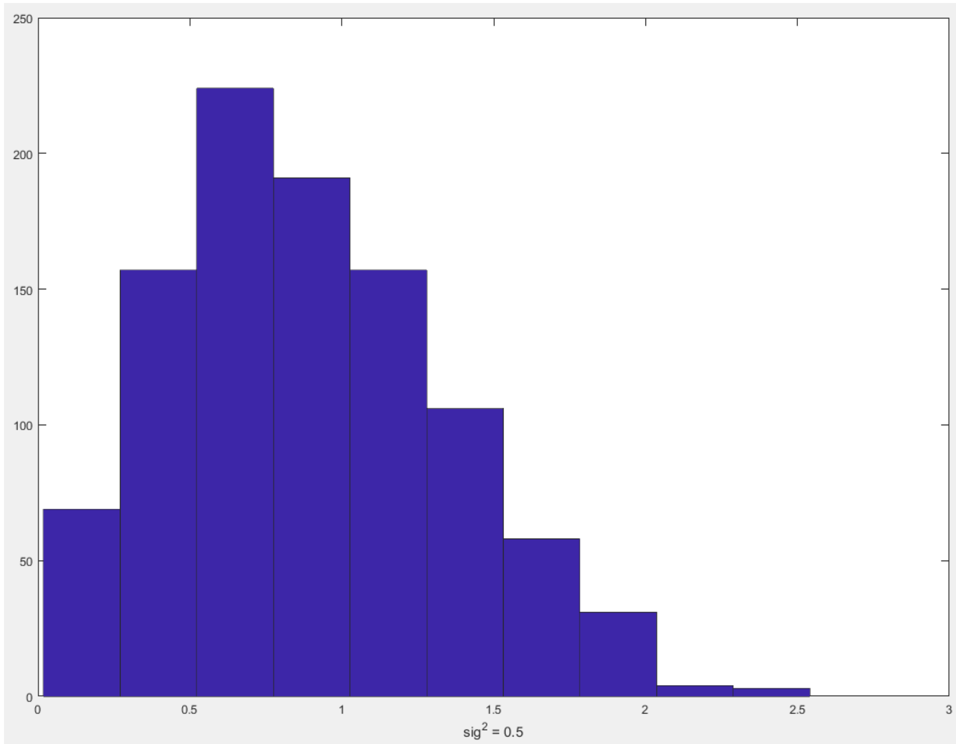
This file can be opened as a Live Script. For more information, see Creating Live Scripts.

```
%%   3.1.c
%%      sig^2 = 0.5
sig = sqrt(0.5);
N=1000;
U=rand(1,N);

x = sqrt(-2*(sig^2)*log(1-U));

samp_mean_x = sum(x)/N
sam_var_x = sum((x-samp_mean_x).^2)/(N-1)

hist(x);
xlabel('sig^2 = 5')

%%       sig^2 = 5
sig = sqrt(5);
N=1000;
U=rand(1,N);

x = sqrt(-2*(sig^2)*log(1-U));

samp_mean_x = sum(x)/N;
sam_var_x = sum((x-samp_mean_x).^2)/(N-1);

hist(x);
xlabel('sig^2 = 5')

%%       sig^2 = 10
sig = sqrt(10);
N=1000;
U=rand(1,N);

x = sqrt(-2*(sig^2)*log(1-U));

samp_mean_x = sum(x)/N;
sam_var_x = sum((x-samp_mean_x).^2)/(N-1);

hist(x);
xlabel('sig^2 = 5')
```

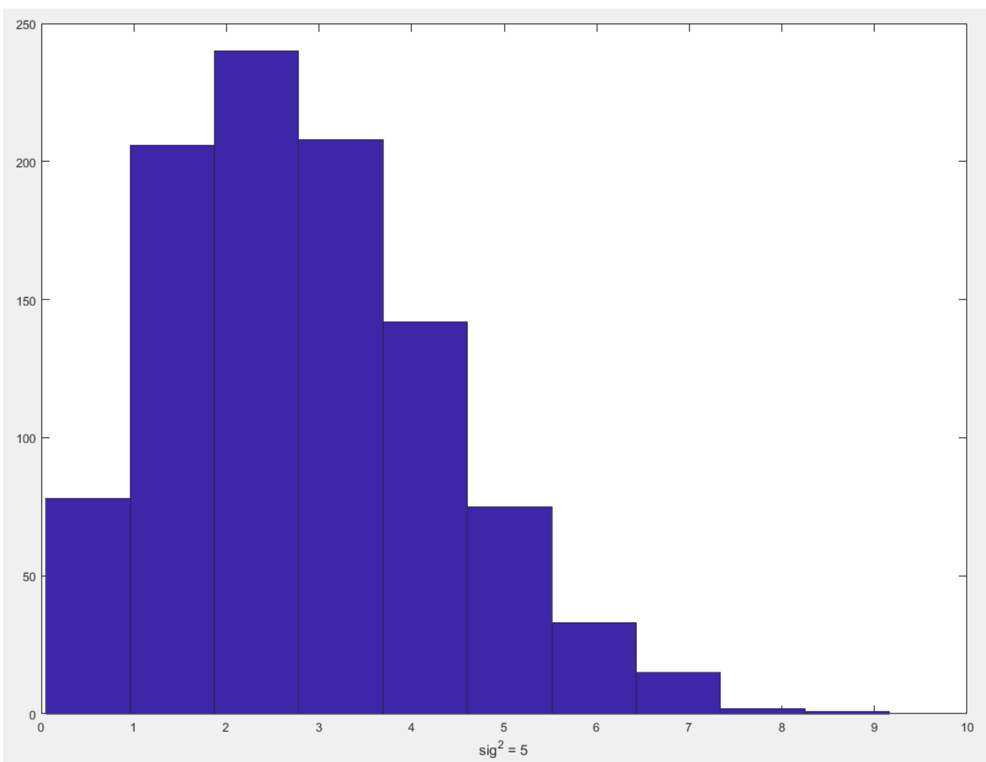
```
samp_mean_x =

    0.8926


sam_var_x =

    0.2160
```

```
samp_mean_x =

    2.7599


sam_var_x =

    2.0299
```

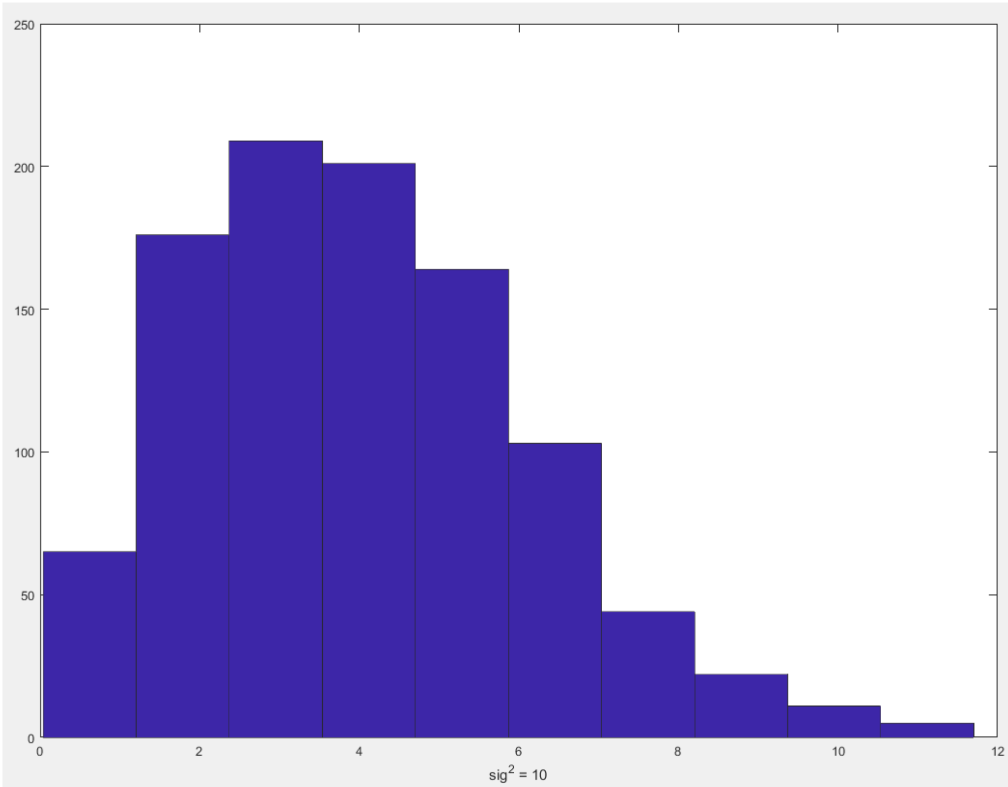
```
samp_mean_x =

    4.0501


sam_var_x =

    4.4028
```



250
200
150
100
50
0
0
0.5
1
1.5
2
2.5
3
sig^2 = 0.5

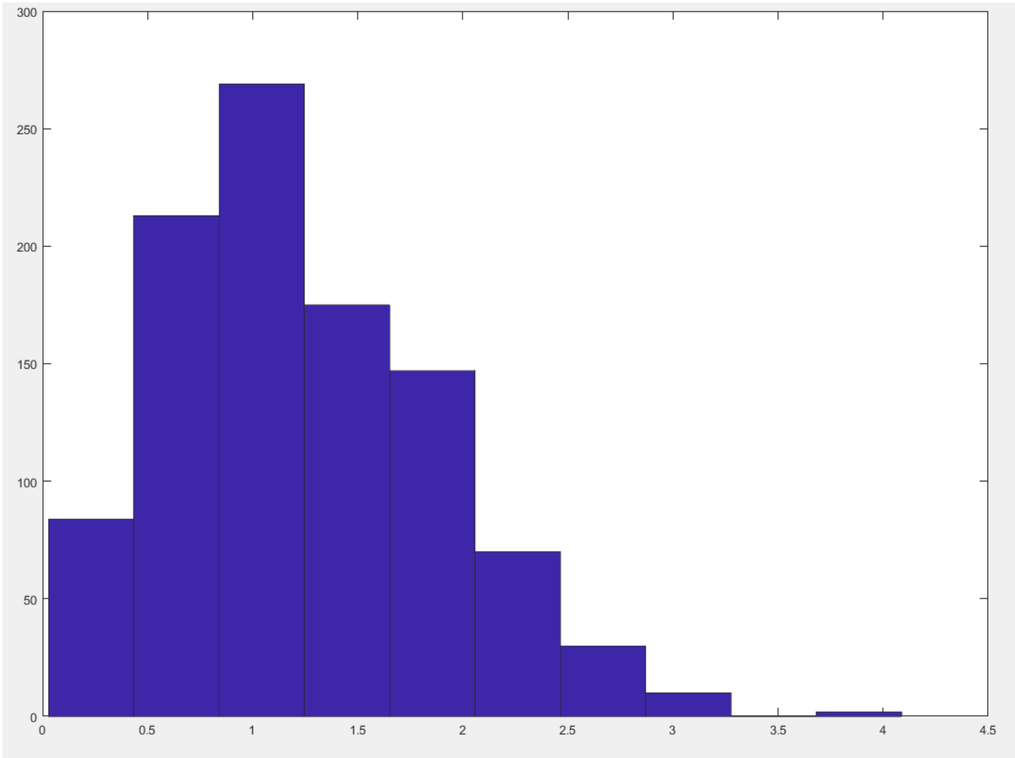


250
200
150
100
50
0
0
1
2
3
4
5
6
7
8
9
10
sig^2 = 5




```
%   3.1.d
clc;
clear;
N=1000;
sig = 1;

Xr = randn(1,N);
Xi = randn(1,N);

%X = Xr + 1i*Xi;
R = sqrt((Xr.^2)+(Xi.^2));

samp_mean_R = sum(R)/N
sam_var_R = sum((R-samp_mean_R).^2)/(N-1)

hist(R);
```

```
samp_mean_R =

    1.2341


sam_var_R =

    0.4067
```



3.1.e

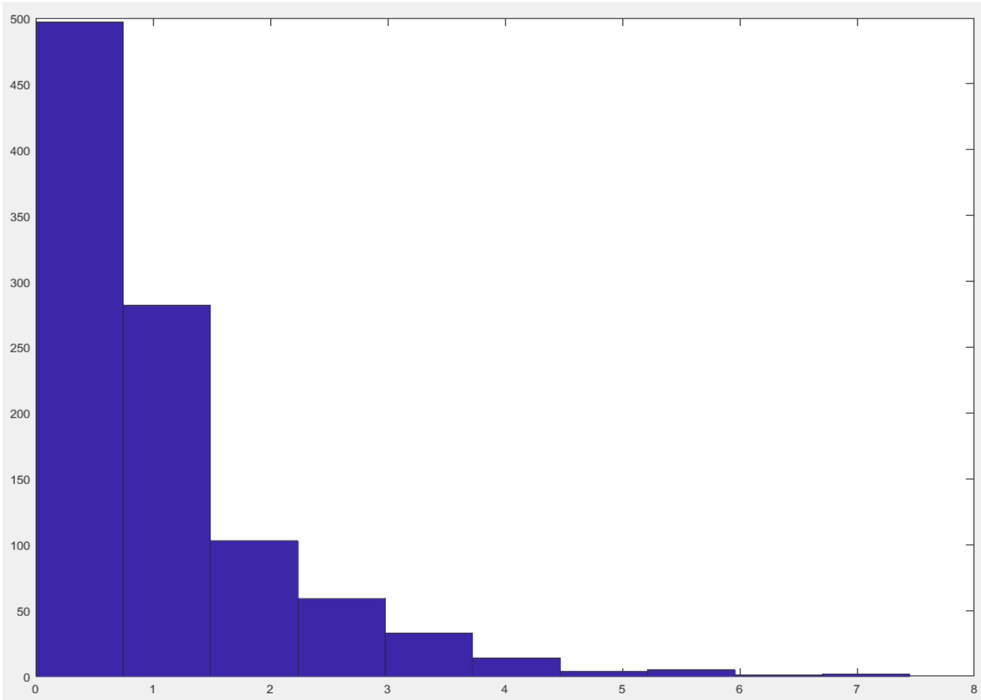
Rayleigh function and Rayleigh distribution from Gaussian samples gives almost same result.

Sample Mean and variance also lies in the same range.

pro_three.m* | Untitled3.m | Untitled4.m | Rayleigh.m

```
%   3.2.a
%       Mean = 1/lambda
%       variance = 1/lambda^2

%   3.2.b

N = 1000;

pdf = exprnd(1,[1,N]);

samp_mean_pdf = sum(pdf)/N
sam_var_pdf = sum((pdf-samp_mean_pdf).^2)/(N-1)

hist(pdf);
```

```
samp_mean_pdf =

    1.0303


sam_var_pdf =

    1.0704
```

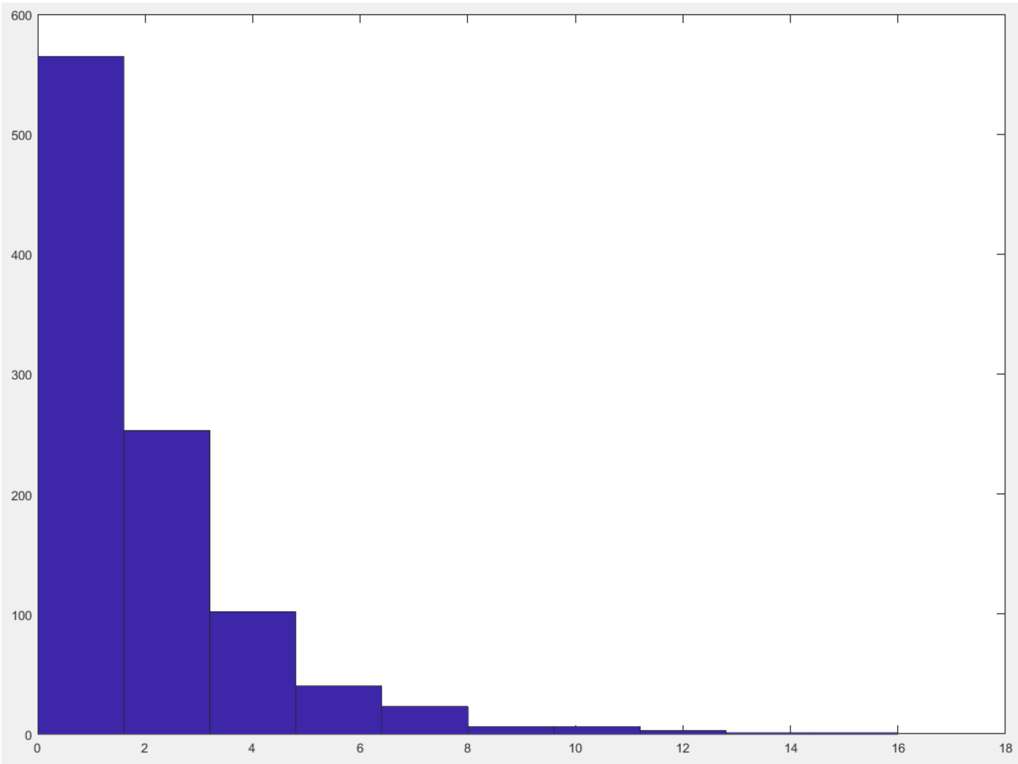


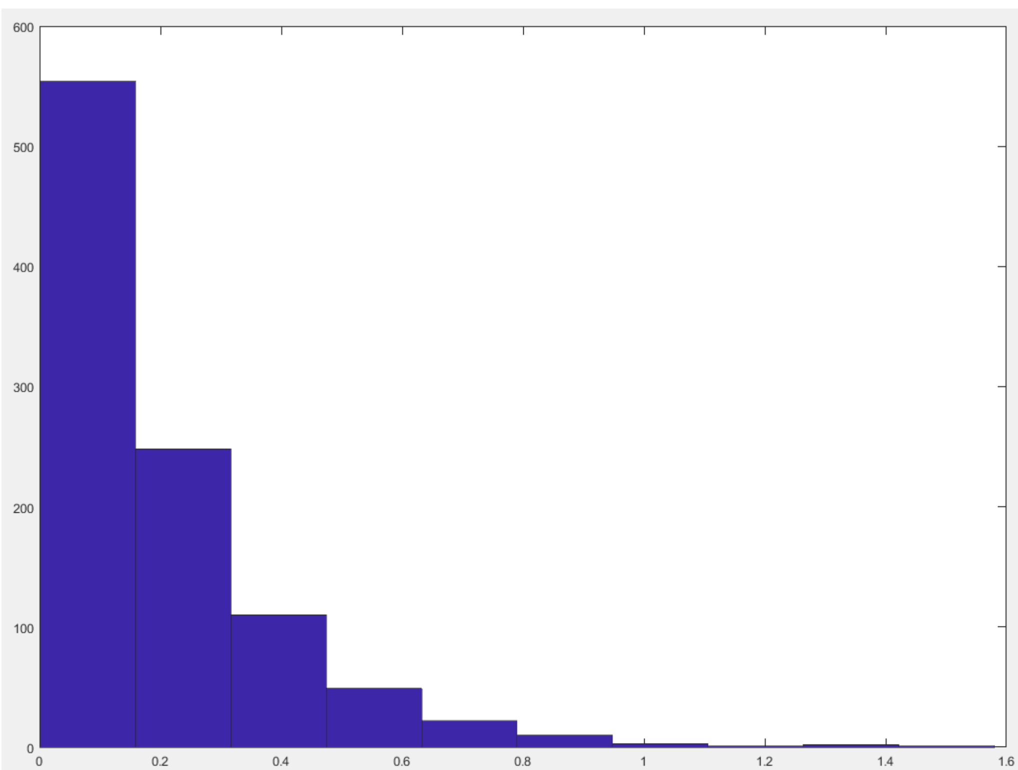

```
%%   3.2.c

N = 1000;
lam = 0.5
mu = 1/lam
pdf = exprnd(mu,[1,N]);

samp_mean_pdf = sum(pdf)/N
sam_var_pdf = sum((pdf-samp_mean_pdf).^2)/(N-1)

hist(pdf);
```

```
samp_mean_pdf =

    1.9326


sam_var_pdf =

    3.9655
```

```
N = 1000;
lam = 5
mu = 1/lam
pdf = exprnd(mu,[1,N]);

samp_mean_pdf = sum(pdf)/N
sam_var_pdf = sum((pdf-samp_mean_pdf).^2)/(N-1)

hist(pdf);
```

```
samp_mean_pdf =

    0.1961


sam_var_pdf =

    0.0385
```



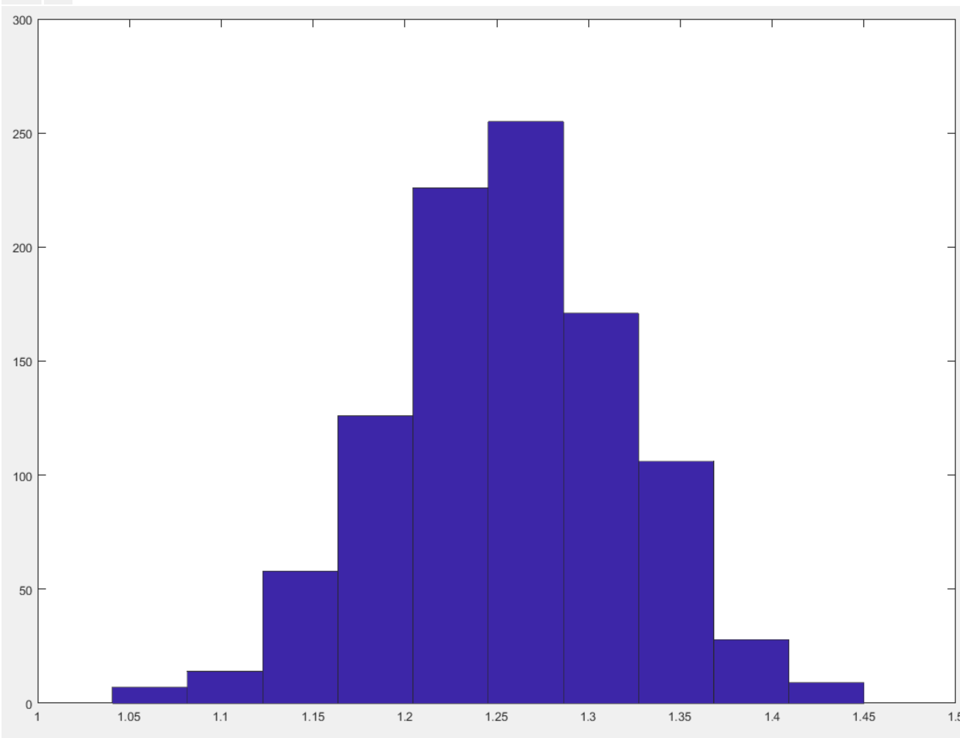


3.2.d: More the λ value small the variance and mean.

4.1.a.

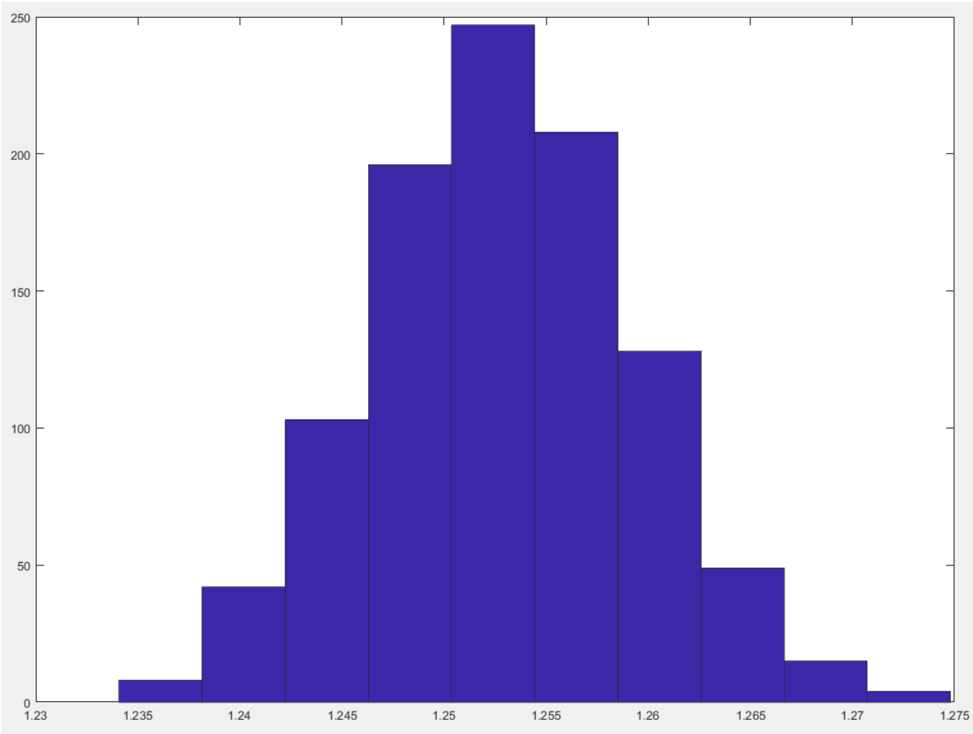
Untitled4.m*

```
for L=1:1000
    sig = 1;
    N=100;
    U=rand(1,N);
    x = sqrt(-2*(sig^2)*log(1-U));
    samp_mean_x(L) = sum(x)/N;
end
hist(samp_mean_x);
```



Yes, from the histogram, it is well approximated to Gaussian pdf.

```
%4.1.b
for L=1:1000
    sig = 1;
    N=10000;
    U=rand(1,N);
    x = sqrt(-2*(sig^2)*log(1-U));
    samp_mean_x(L) = sum(x)/N;
end
hist(samp_mean_x);
```

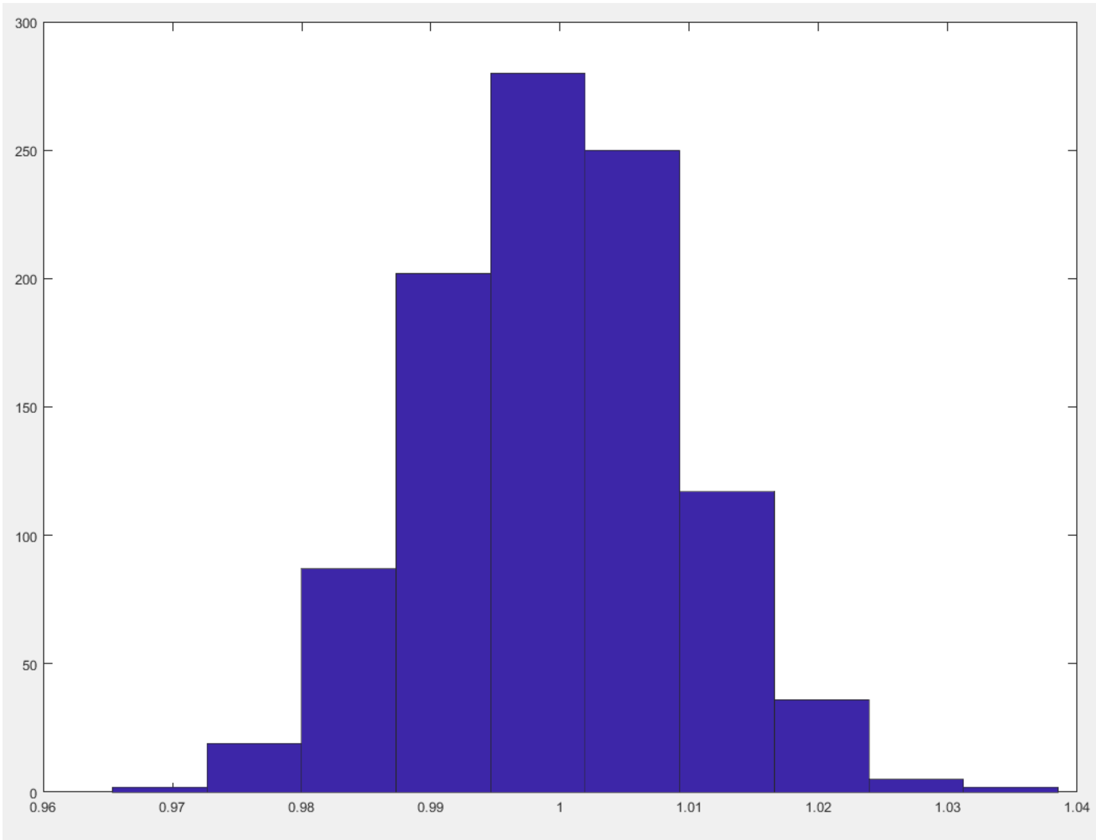


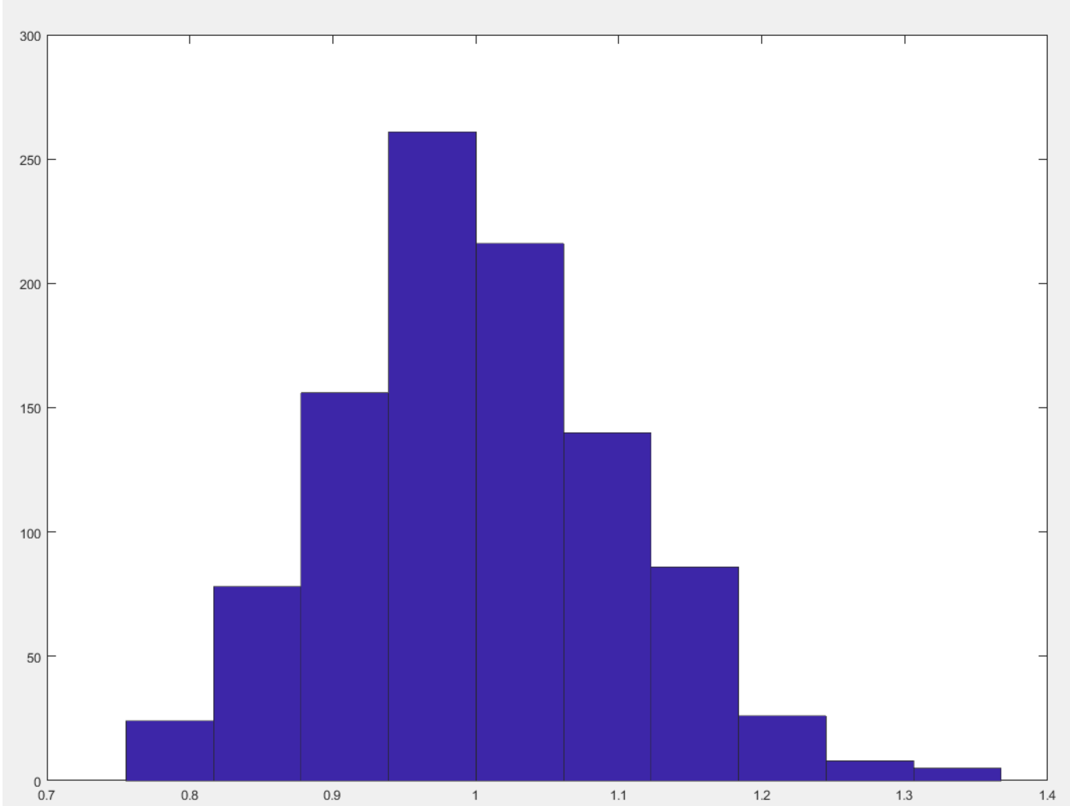
Yes, I think this can also be well approximated to Gaussian pdf.

When compared with the above one this can be approximated more.

```
%% 4.1.c
for L=1:1000
    N = 100;
    lam = 1;
    mu = 1/lam;
    y = exprnd(mu,[1,N]);
    samp_mean_y(L) = sum(y)/N;
end

hist(samp_mean_y);
```

```
%% 4.1.d
for L=1:1000
    N = 10000;
    lam = 1;
    mu = 1/lam;
    y = exprnd(mu,[1,N]);
    samp_mean_y(L) = sum(y)/N;
end

hist(samp_mean_y);
```


300
250
200
150
100
50
0
0.96
0.97
0.98
0.99
1
1.01
1.02
1.03
1.04


300
250
200
150
100
50
0
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4

```
%4.1.e
for L=1:1000
    sig = 1;
    N=100;
    U=rand(1,N);
    x = sqrt(-2*(sig^2)*log(1-U));
    samp_mean_x(L) = sum(x)/N;
end

for L=1:1000
    N = 100;
    lam = 1;
    mu = 1/lam;
    y = exprnd(mu,[1,N]);
    samp_mean_y(L) = sum(y)/N;
end

z = sum(x)/sum(y)

%4.1.f
for L=1:1000
    sig = 1;
    N=10000;
    U=rand(1,N);
    x = sqrt(-2*(sig^2)*log(1-U));
    samp_mean_x(L) = sum(x)/N;
end

for L=1:1000
    N = 10000;
    lam = 1;
    mu = 1/lam;
    y = exprnd(mu,[1,N]);
    samp_mean_y(L) = sum(y)/N;
end

z = sum(x)/sum(y)
```

```
z =

    1.1567


z =

    1.2261
```