

1

```
1 - clc;
2 - clear;
3 - % 1.a
4 - x=binornd(1,0.6,[1,20])
5 - % 1.b
6 - hist(x);
7 - % Not always showing the output exactly to the probability of "p".
8 - % Histogram plot plots number of occurrences of the same value.
```

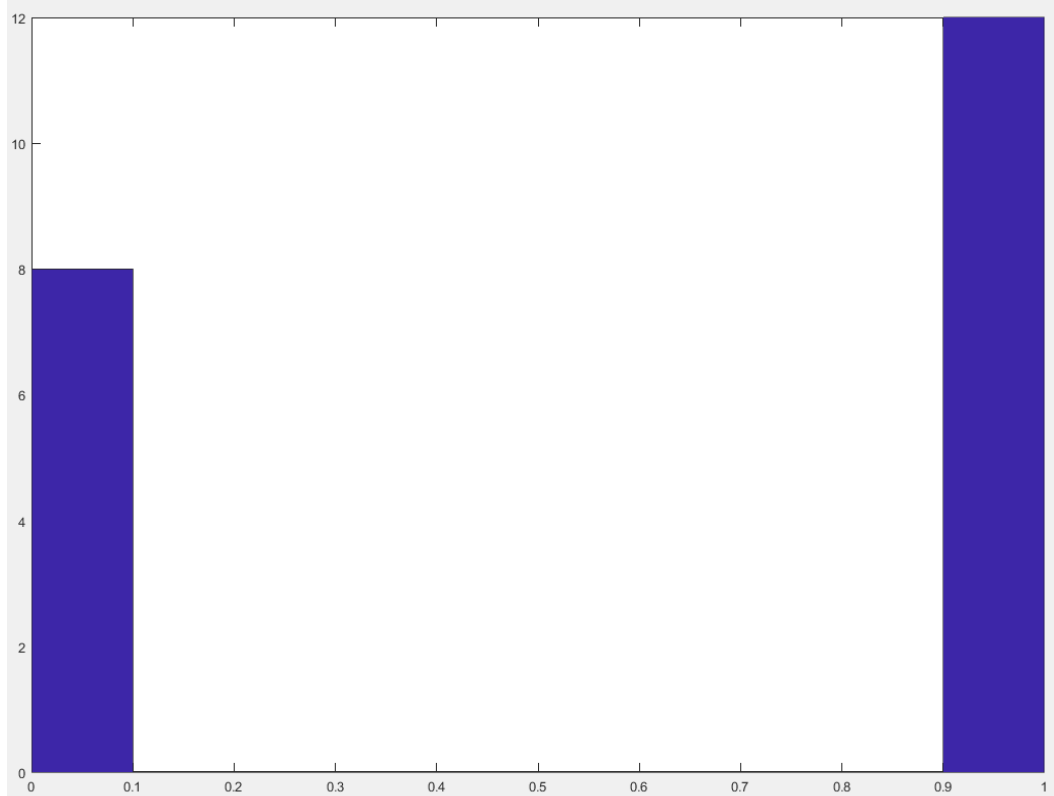
x =

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  % 1.c
2  function x= my_bernoulli(p,N,a,b);
3
4  x=rand(1,N);
5
6  if or(a,b)==0
7      if a>b
8          x(x<(1-p))=b;
9          x(x>=(1-p))=a;
10         x
11         hist(x);
12     elseif a<b
13         x(x<(p))=a;
14         x(x>=(p))=b;
15         x
16         hist(x);
17     else
18         display('Not a bernoulli Random variable');
19     end
20
21 else
22     if a>b
23         x(x>=(1-p))=a;
24         x(x<(1-p))=b;
25         x
26         hist(x);
27     elseif a<b
28         x(x>=(p))=b;
29         x(x<(p))=a;
30         x
31         hist(x);
32     else
33         display('Not a bernoulli Random variable');
34     end
35
36 end
37
```

```

my_bernoulli.m  Main.m*  +
1 -  clc;
2 -  clear;
3
4 -  x=my_bernoulli(0.6,20,1,0);
5
6 -  %      Analytical mean = mx
7 -  %      Analytical Variance = Vx
8
9 -  %      2.1.a
10 - %      mx = p = 0.6
11 - %      vx = p(1-p) = 0.6*0.4 = 0.24
12
13 - mx=0.6;
14 - vx=0.24;
15
16 - %      System mean = mean_x
17 - %      System Variance = var_x
18
19 - mean_x = mean(x);
20 - var_x = var(x);
21
22 - %      Sample Mean = sam_mean_x
23 - %      Sample Variance = sam_var_x
24
25 - N=length(x)
26 - sam_mean_x = sum(x)/N;
27 - sam_var_x = sum((x-sam_mean_x).^2)/(N-1);
28
29
30
31 - z=[mx vx mean_x var_x  sam_mean_x  sam_var_x]
32 - display('      mx      vx      mean(x)  var(x)  sample mean(x)  sample var(x)')
33
34 - figure
35 - bar(z)
36 - xlabel('mx vx mean(x) var(x) sample mean(x) sample var(x)')
37 - ylabel('Value')
38
39 - % The sample mean and variance are same as the matlab function mean and
40 - % 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);

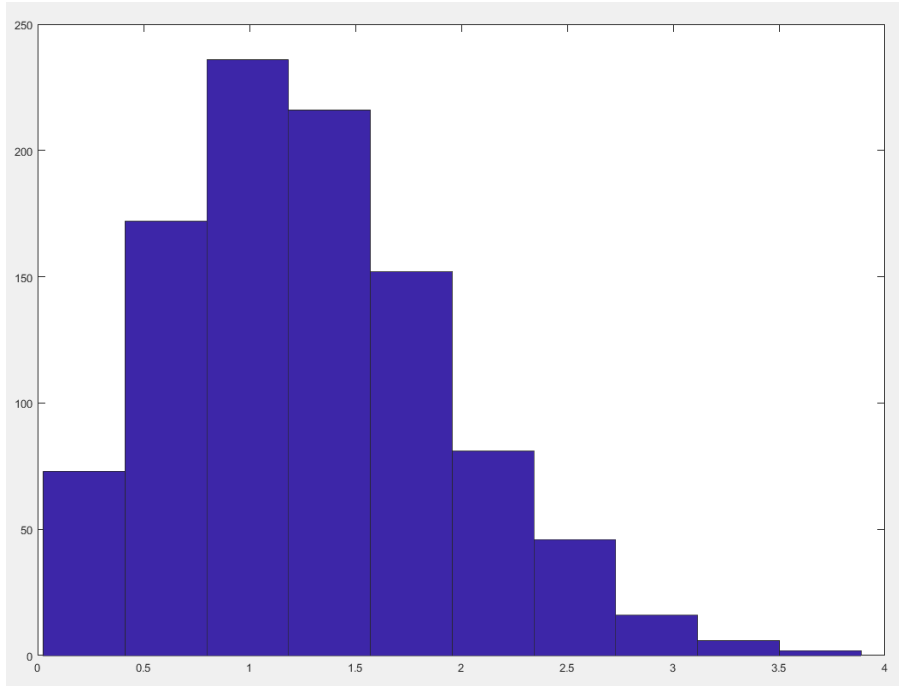
samp_mean_x =

    1.2840

sam_var_x =

    0.4246

```



```

Rayleigh.m*  x  +
This file can be opened as a Live Script. For more information, see Creating Live Scripts.

26  %% 3.1.c
27  %% sig^2 = 0.5
28  sig = sqrt(0.5);
29  N=1000;
30  U=rand(1,N);
31
32  x = sqrt(-2*(sig^2)*log(1-U));
33
34  samp_mean_x = sum(x)/N
35  sam_var_x = sum((x-samp_mean_x).^2)/(N-1)
36
37  hist(x);
38  xlabel('sig^2 = 5')
39
40  %% sig^2 = 5
41  sig = sqrt(5);
42  N=1000;
43  U=rand(1,N);
44
45  x = sqrt(-2*(sig^2)*log(1-U));
46
47  samp_mean_x = sum(x)/N;
48  sam_var_x = sum((x-samp_mean_x).^2)/(N-1);
49
50  hist(x);
51  xlabel('sig^2 = 5')
52
53  %% sig^2 = 10
54  sig = sqrt(10);
55  N=1000;
56  U=rand(1,N);
57
58  x = sqrt(-2*(sig^2)*log(1-U));
59
60  samp_mean_x = sum(x)/N;
61  sam_var_x = sum((x-samp_mean_x).^2)/(N-1);
62
63  hist(x);
64  xlabel('sig^2 = 5')
65

```

```

samp_mean_x =

    0.8926

sam_var_x =

    0.2160

```

```

samp_mean_x =

    2.7599

sam_var_x =

    2.0299

```

```

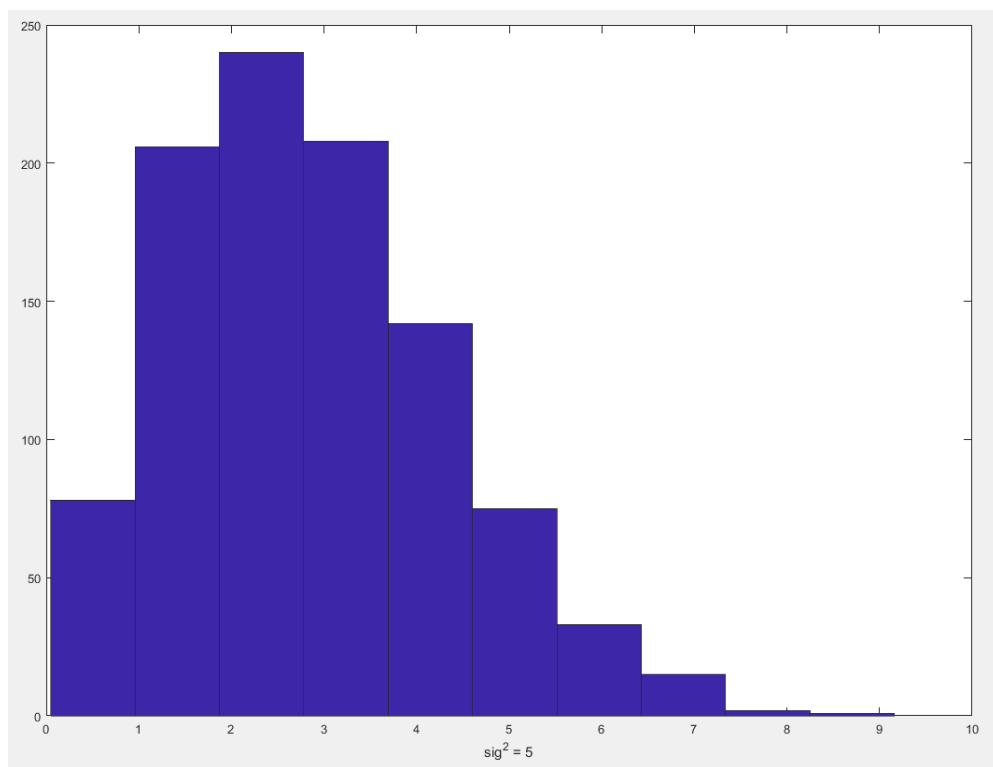
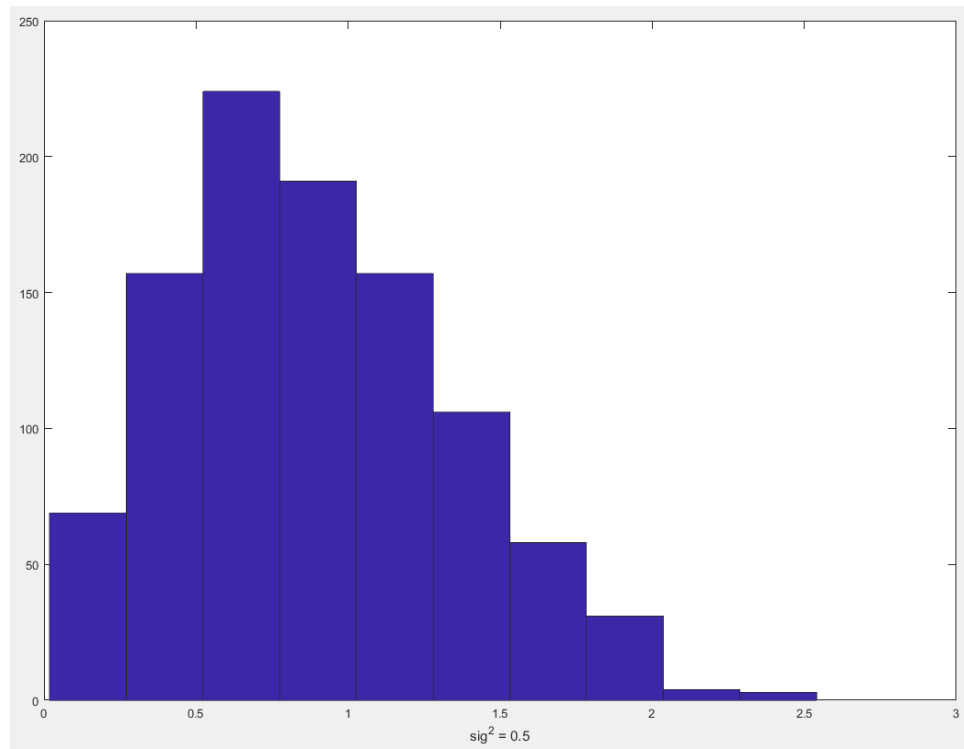
samp_mean_x =

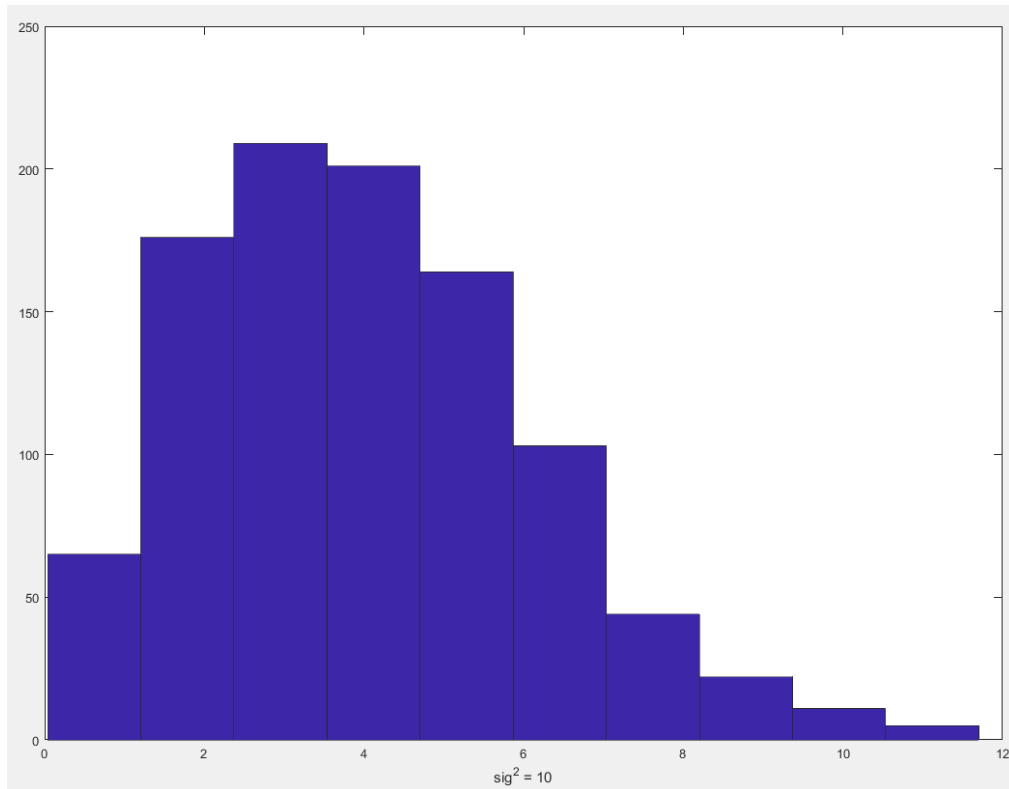
    4.0501

sam_var_x =

    4.4028

```





```

1  % 3.1.d
2  - clc;
3  - clear;
4  - N=1000;
5  - sig = 1;
6
7  - Xr = randn(1,N);
8  - Xi = randn(1,N);
9
10 %X = Xr + 1i*Xi;
11 - R = sqrt((Xr.^2)+(Xi.^2));
12
13 - samp_mean_R = sum(R)/N
14 - sam_var_R = sum((R-samp_mean_R).^2)/(N-1)
15
16 - hist(R);

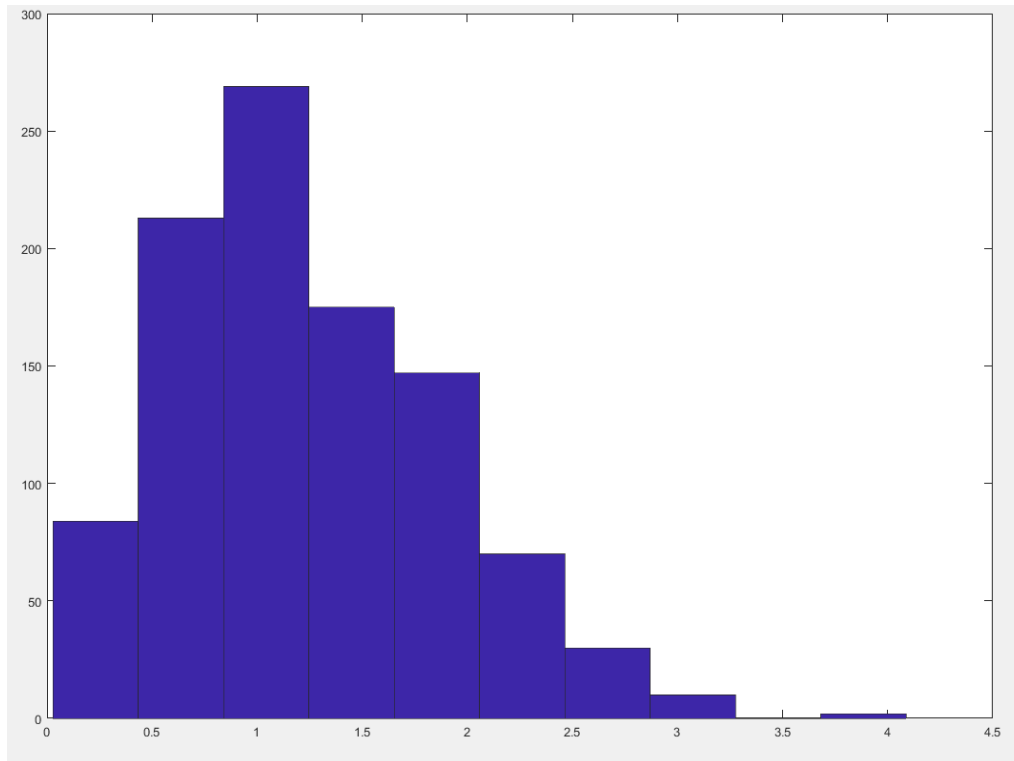
```

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.

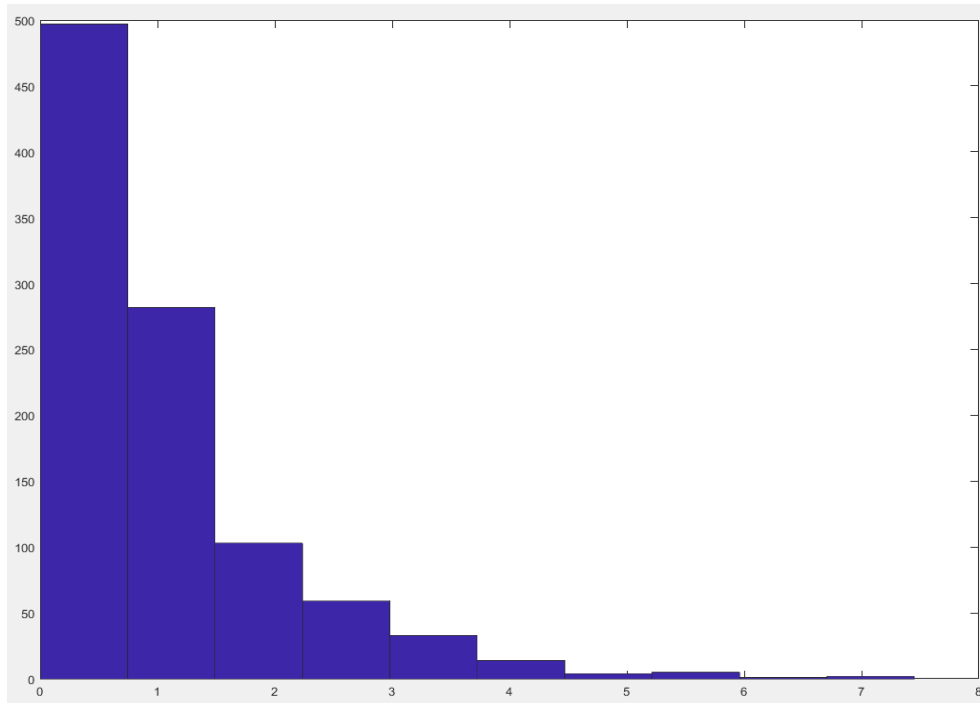
Sample Mean and variance also lies in the same range.

```

pro_three.m*  Untitled3.m  Untitled4.m  Rayleigh.m  +
1      % 3.2.a                                samp_mean_pdf =
2      %      Mean = 1/lambda
3      %      variance = 1/lambda^2                1.0303
4
5      % 3.2.b                                sam_var_pdf =
6
7      N = 1000;                                1.0704
8
9      pdf = exprnd(1,[1,N]);
10
11      samp_mean_pdf = sum(pdf)/N
12      sam_var_pdf = sum((pdf-samp_mean_pdf).^2)/(N-1)
13
14      hist(pdf);
15
16

```





```
%% 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);
```

```
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)

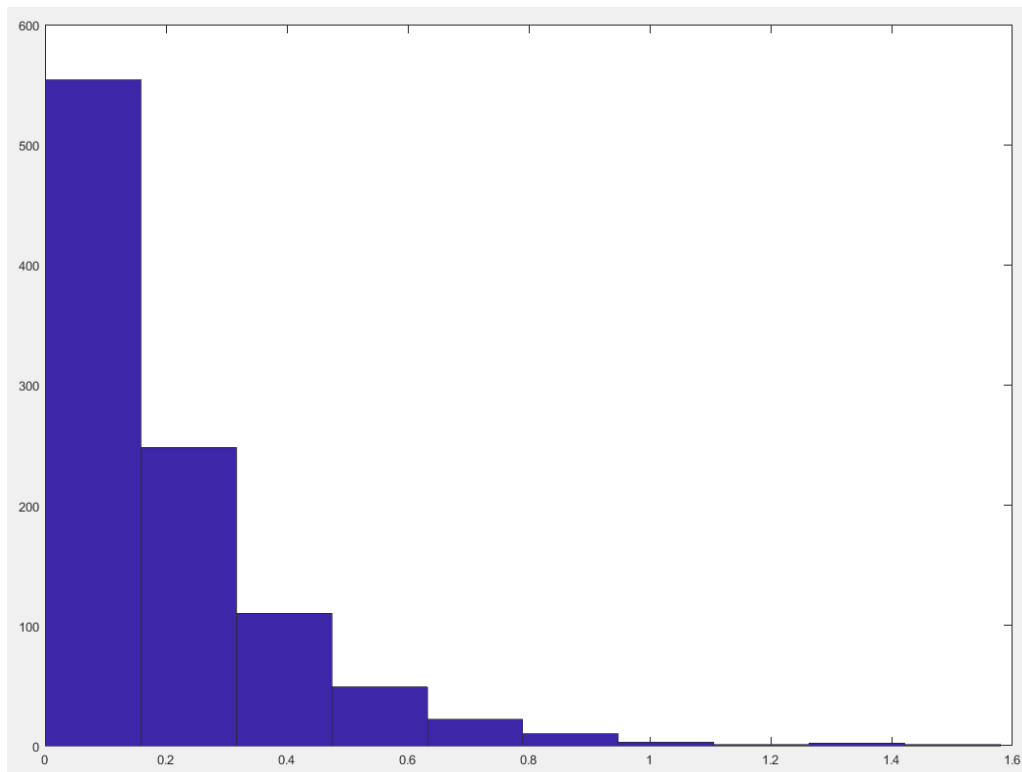
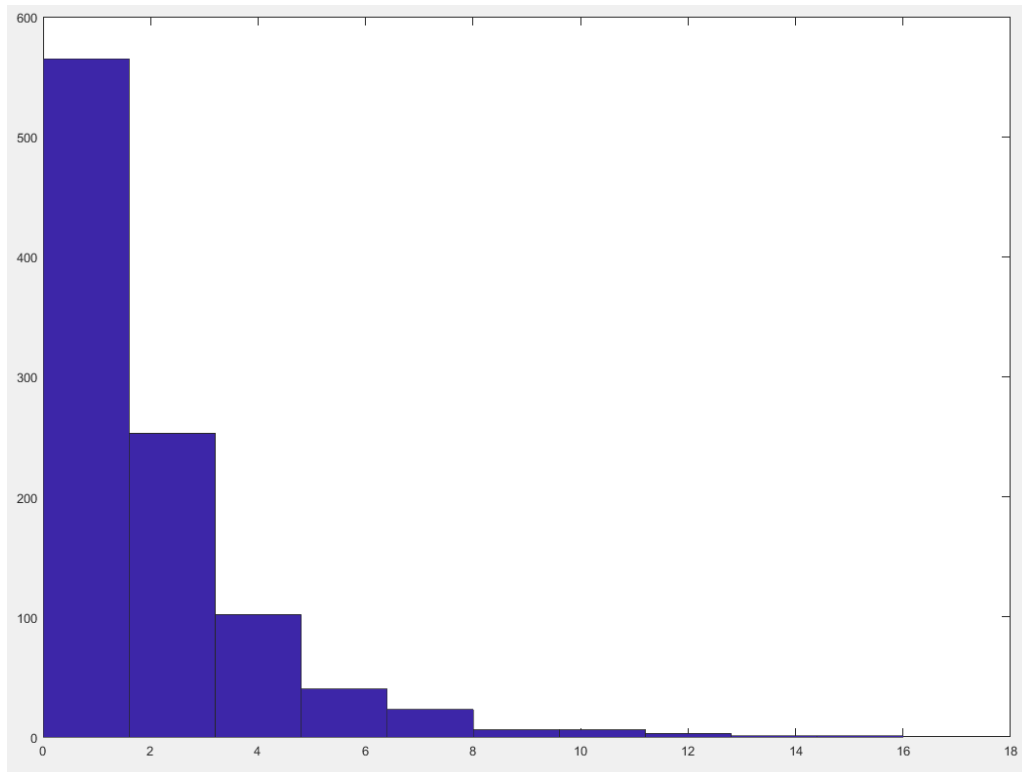
hist(pdf);
```

```
samp_mean_pdf =
    1.9326

sam_var_pdf =
    3.9655
```

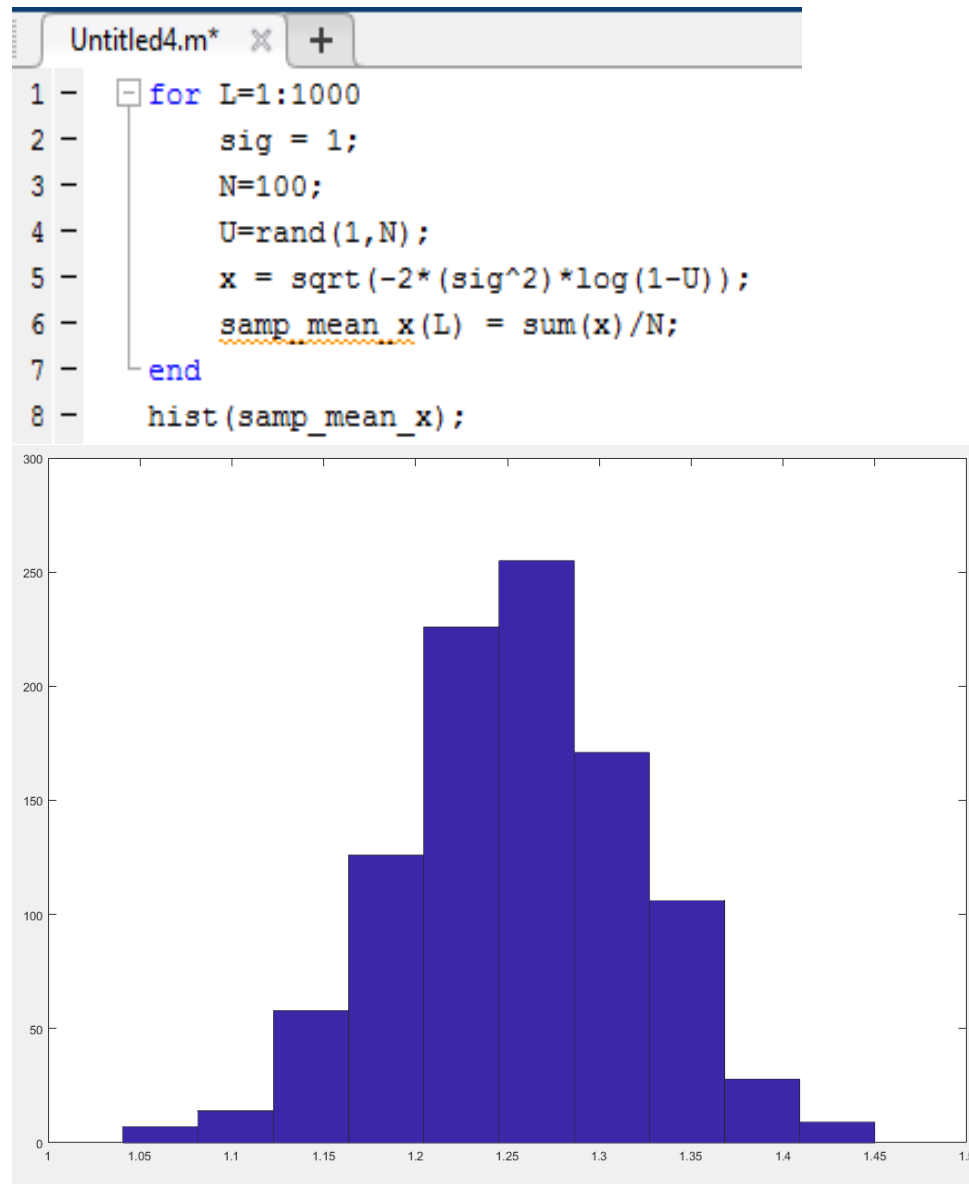
```
samp_mean_pdf =
    0.1961

sam_var_pdf =
    0.0385
```



3.2.d: More the  $\lambda$  value small the variance and mean.

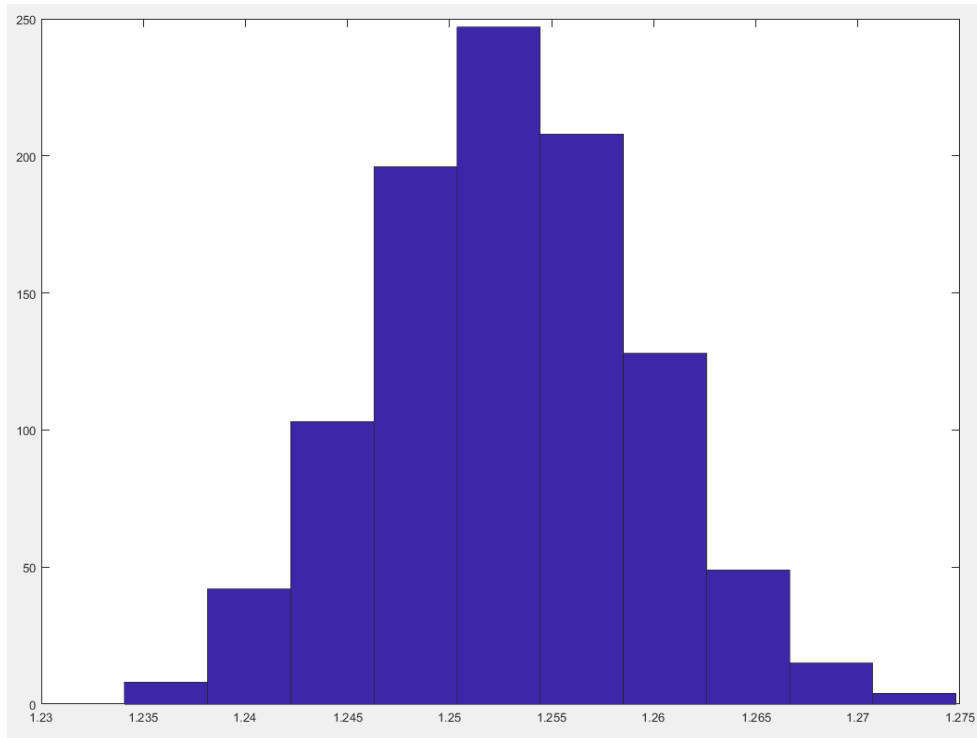
4.1.a.



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

%4.1.b

```
1 - for L=1:1000
2 -     sig = 1;
3 -     N=10000;
4 -     U=rand(1,N);
5 -     x = sqrt(-2*(sig^2)*log(1-U));
6 -     samp_mean_x(L) = sum(x)/N;
7 - end
8 - hist(samp_mean_x);
```



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

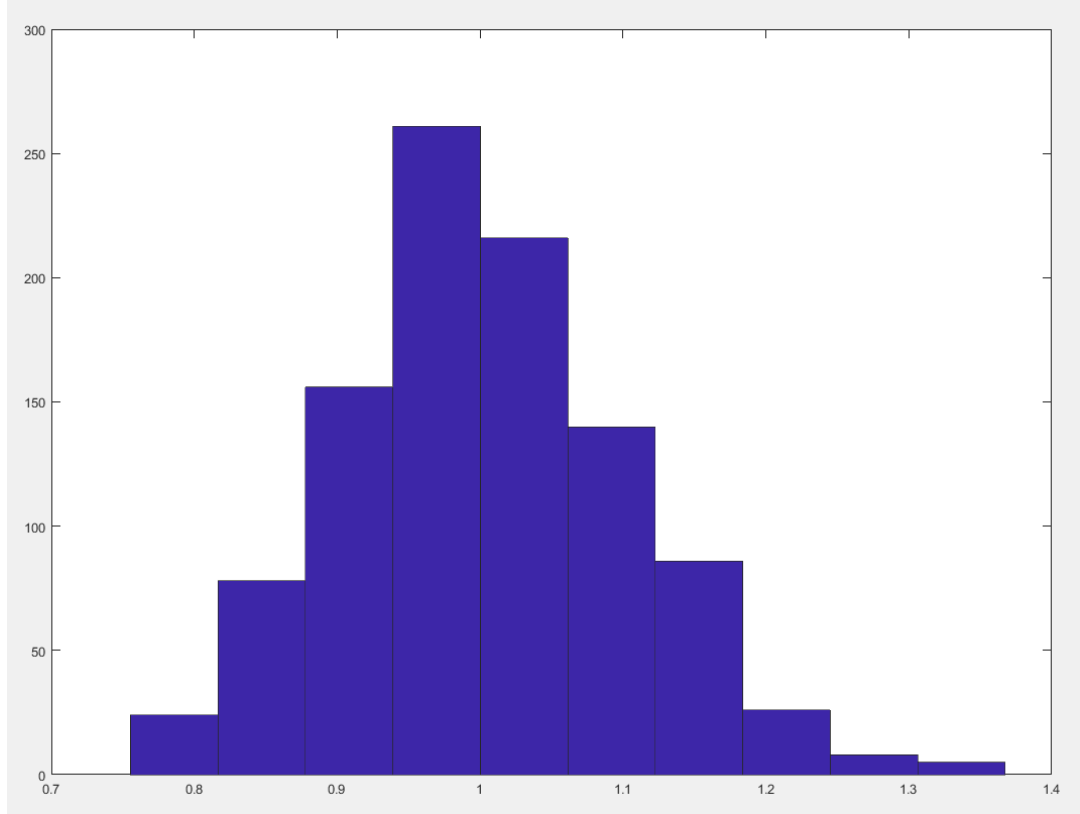
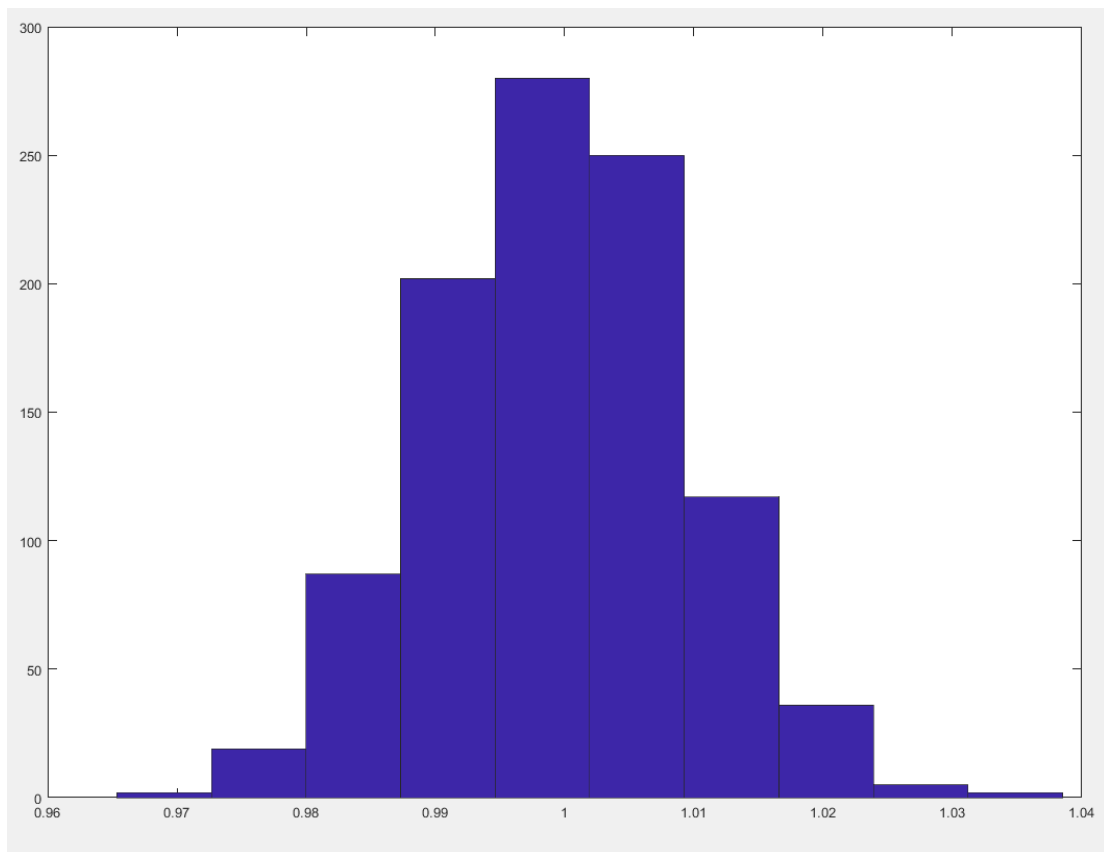
When compared with the above one this can be approximated more.

```
%% 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);
```



```

1      %4.1.e
2  -   for L=1:1000
3  -       sig = 1;
4  -       N=100;
5  -       U=rand(1,N);
6  -       x = sqrt(-2*(sig^2)*log(1-U));
7  -       samp_mean_x(L) = sum(x)/N;
8  -   end
9
10  -   for L=1:1000
11  -       N = 100;
12  -       lam = 1;
13  -       mu = 1/lam;
14  -       y = exprnd(mu,[1,N]);
15  -       samp_mean_y(L) = sum(y)/N;
16  -   end
17
18  -   z = sum(x)/sum(y)
19
20      %4.1.f
21  -   for L=1:1000
22  -       sig = 1;
23  -       N=10000;
24  -       U=rand(1,N);
25  -       x = sqrt(-2*(sig^2)*log(1-U));
26  -       samp_mean_x(L) = sum(x)/N;
27  -   end
28
29  -   for L=1:1000
30  -       N = 10000;
31  -       lam = 1;
32  -       mu = 1/lam;
33  -       y = exprnd(mu,[1,N]);
34  -       samp_mean_y(L) = sum(y)/N;
35  -   end
36
37  -   z = sum(x)/sum(y)
38
39

```

z =

1.1567

z =

1.2261