

Statistical Techniques for Analyzing a Continuous Distribution

1.1 Objective of the Project

The objective of this project is to define a probability density function, and to derive the Cumulative Distribution Function – by two methods; To calculate sample mean, sample variance, population mean, population variance, and mean square estimates from all these parameters; To supplement this by an example; To implement the concept of bootstrapping and comparing the values.

1.2 Problem Statement

Consider a continuous Random Variable X defined by its probability density function as

$$f(x) = C * \sin(x), 0 \leq x \leq \pi \text{ and } 0 \text{ otherwise}$$

Where C is a normalizing constant.

Part A – To find the value of C , to find the Cumulative Distribution Function. To compute the population, mean and population variance.

Part B – To generate the actual samples X by taking the inverse of the distribution function. Generate three sets of samples – for $N = 25$, $N = 100$ and $N = 1000$. To compare the distribution plots.

Part C – For $N=100$ samples, to find the sample mean and population variance. To find MSE using population variance.

Assuming population variance is unknown, to compute MSE using sample variance.

To compare the obtained results.

Part D – For $N=100$, to use bootstrapping technique to generate 50 bootstrap samples. Extend to 100, 1000 and 10000 samples. Computing mean and variance.

Part E – To find the MSE of the generated bootstrap sample in Part D

1.3 Mathematical Basis

For a continuous pdf $f(x)$, the CDF can be generated as

$$F(x) = \int_{-\infty}^x f(x) dx$$

In Matlab, every continuous function is implemented as a sum of large numbers of discrete functions. Hence mean and variance are given for the discrete case. Considering the set of iid random variables X_1, X_2, \dots, X_n , each of whose variance is defined, Sample Mean and Sample Variance are obtained per sample. It is calculated as

$$\text{Sample Mean } \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

$$\text{Sample Variance} = \frac{1}{(n-1)} \sum_{i=1}^n (X_i - \bar{X})^2$$

The population Mean and variance in the interval $[0, \pi]$ is given in the continuous case as

$$E[X] = \int_0^{\pi} x * f(x) dx$$

$$\text{Var}(X) = E[X^2] - (E[X])^2 = \int_0^{\pi} x^2 * f(x) dx - \left(\int_0^{\pi} x * f(x) dx \right)^2$$

To know how good the sample mean is the estimator of the population mean, the parameter MSE – Mean Square Error is used. The MSE of mean is given as

$$\text{MSE}(\bar{X}) = E\left[(\bar{X} - \mu)^2\right] = \left(\frac{\sigma}{\sqrt{n}}\right)^2 = \frac{\sigma^2}{n}$$

MSE is a measure of how close the computed mean is to the actual mean. A large value of MSE indicated more deviation.

Bootstrapping is just the re – creation of samples to create a large, non-repeated, samples from a population to define a sampling distribution for a statistic. Bootstrapping is a good method for computing the population parameters. Bootstrapping uses empirical distribution to define the discrete sample space distribution for each X_i . The samples are drawn from the empirical distribution.

The variance of the empirical distribution is given as

$$\sigma_{F^*}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - m_{F^*})^2 \text{ where } m_{F^*} = \frac{1}{n} \sum_{i=1}^n x_i$$

Hence sample mean and sample variance can be calculated for each of the bootstrap samples. The Mean Square Estimate of the Variance is also computed as

$$\text{MSE}(\sigma_{F^*}^2) = \frac{1}{k} \sum_{j=1}^k (S_j^2 - \sigma_{F^*}^2)^2$$

1.4 Simulation in Matlab

The given problem is simulated in Matlab as follows.

1.4.1 Description of the code

The entire project is compiled as a single program. The pdf equation is given as the input, with an unknown constant C.

To find the value of C, the pdf is integrated over the range $[0, \pi]$. Then the obtained C value is substituted in the pdf equation. Now, the pdf is in the closed form. The Cumulative Distribution Function is obtained by integrating the pdf over the range $[0, x]$. The two arrays pdf_val and cdf_val give the value of the respective functions evaluated at points 0.01 apart in the interval $[0, \pi]$.

Mean of the distribution is evaluated in the interval $[0, \pi]$ using the formula

$$E[X] = \int_0^{\pi} x * f(x) dx$$

Variance of the distribution is evaluated in the interval $[0, \pi]$ using the formula

$$\text{Var}(X) = E[X^2] - (E[X])^2 = \int_0^{\pi} x^2 * f(x) dx - \left(\int_0^{\pi} x * f(x) dx \right)^2$$

The inverse of the distribution function is obtained using the command finverse.

For N=25 samples, the random values are generated using the rand command. Each of the generated random value is substituted in the obtained inverse function. This cdf is plotted and compared with the actual cdf.

This is repeated for N=100 and N=1000 samples.

For the case of N=100, the sample mean and population variance is found. Using this MSE of the mean is estimated. It is again assumed that the population variance is unknown, so sample variance is computed using the formula, and the values are compared.

For the case of N=100, bootstrapping is implemented using the command bootstrp for 50 samples. Mean and sample variance are calculated and compared. Finally, MSE for the generated bootstrap samples is also calculated.

1.4.2 Observations

The continuous pdf is given by

$$f(x) = C * \sin(x), 0 \leq x \leq \pi \text{ and } 0 \text{ otherwise}$$

This function is integrated to find the value of C as in Figure 1. The c – substituted pdf equation is also shown in Figure 1.

```

Command Window

Normalizing constant :

c =

1/2

pdf_func =

sin(x)/2

```

Figure 1 Value of the normalizing constant

The Cumulative distribution function obtained by integrating pdf is shown in Figure 2.

```

Command Window

CDF :

1/2 - cos(x)/2

```

Figure 2 CDF of the function.

A sample of values for the CDF for some x values is shown in Figure 3. The formula used for computation is right, since there is a non – decreasing sequence of values obtained in Figure 3.

```

Command Window

cdf_val =

Columns 1 through 6

0.0032    0.1302    0.2635    0.4127    0.5905    1.0000

Columns 7 through 11

1.0000    1.0000    1.0000    1.0000    1.0000

```

Figure 3 A sample of a small set of CDF values.

The actual mean and variance of the distribution is obtained using the formula as in Figure 4.

```

Command Window

mean_val =

1.5708

e_x2 =

2.9348

variance_x =

0.4674

```

Figure 4 Mean and Variance of the distribution

The actual values of the distribution computed using the inbuilt mean and var command is shown in Figure 5.

```
Command Window  
  
mean_analy =  
  
    1.5698  
  
>> var_analy  
  
var_analy =  
  
    0.4621
```

Figure 5 Analytical Mean and Variance Values

The plots of PDF and CDF are shown in Figure 6(a) and 6(b) respectively.

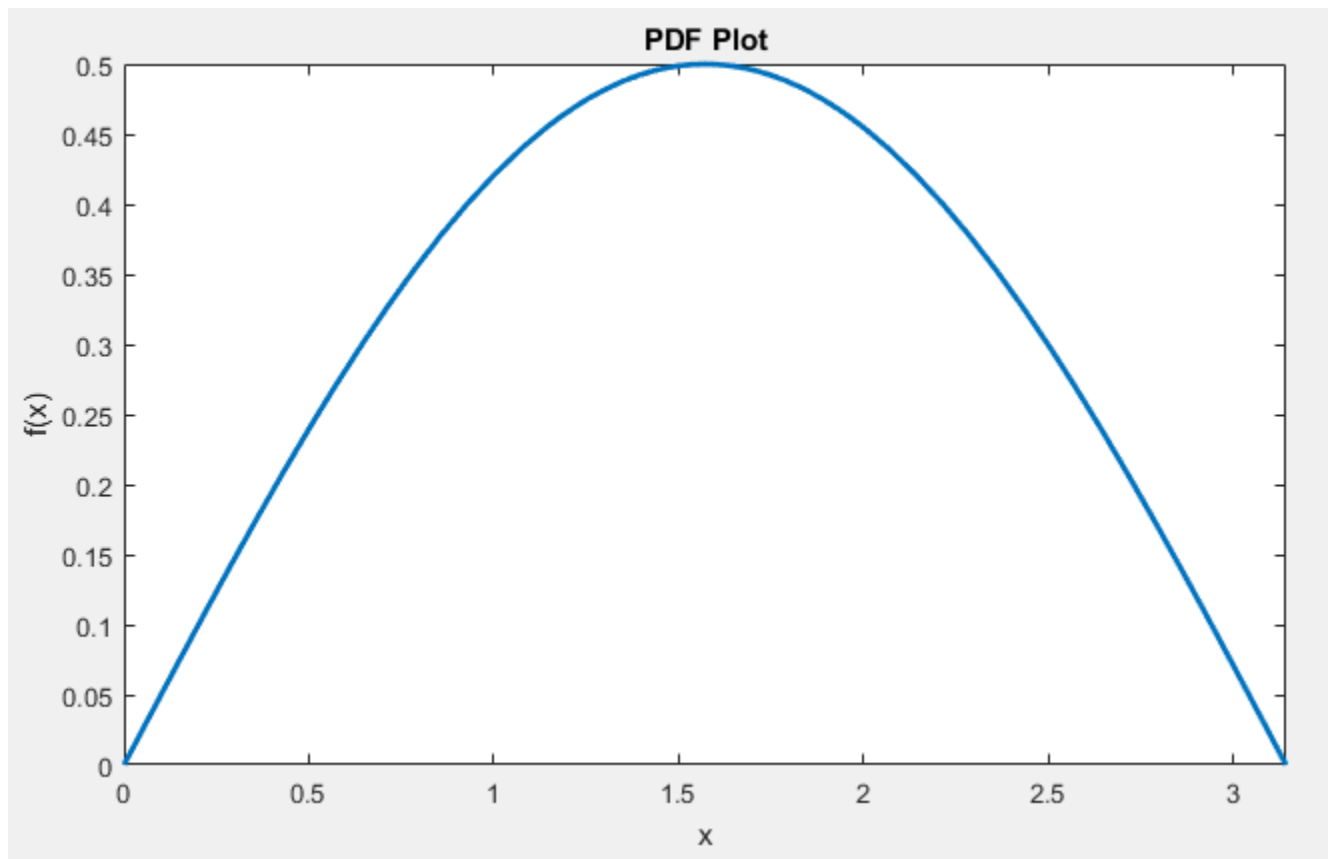


Figure 6(a) PDF Plot

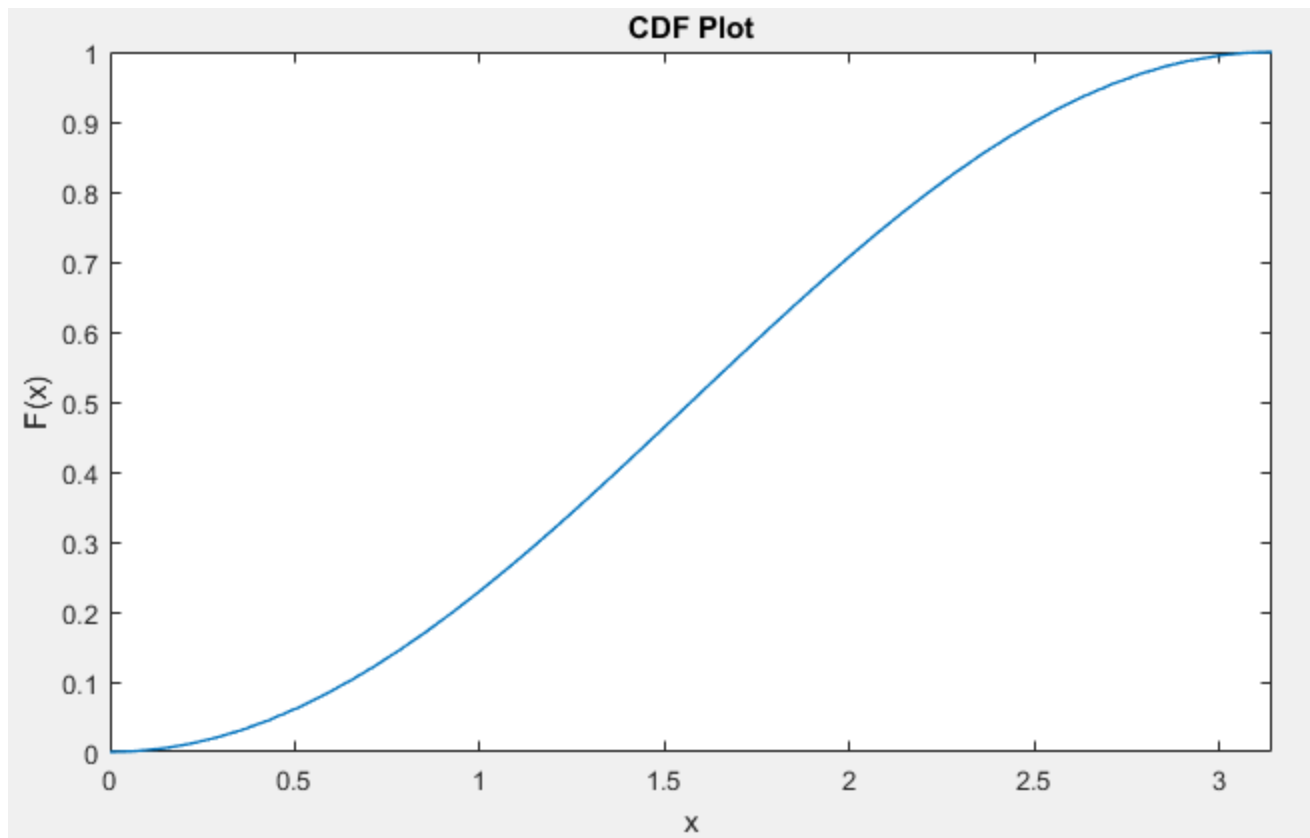


Figure 6(b) CDF Plot

The obtained CDF is inverted using the finverse command. The obtained inverse CDF is shown in Figure 7.

```
Command Window

Inverse of CDF:

inv_func =

acos(1 - 2*x)
```

Figure 7 Inverse CDF

Then $N=25$ samples are generated using the rand command. Its CDF is found by substituting the obtained values in the inverse CDF equation. Figure 8 shows the CDF plot comparing the CDF of the actual distribution and obtained distribution.

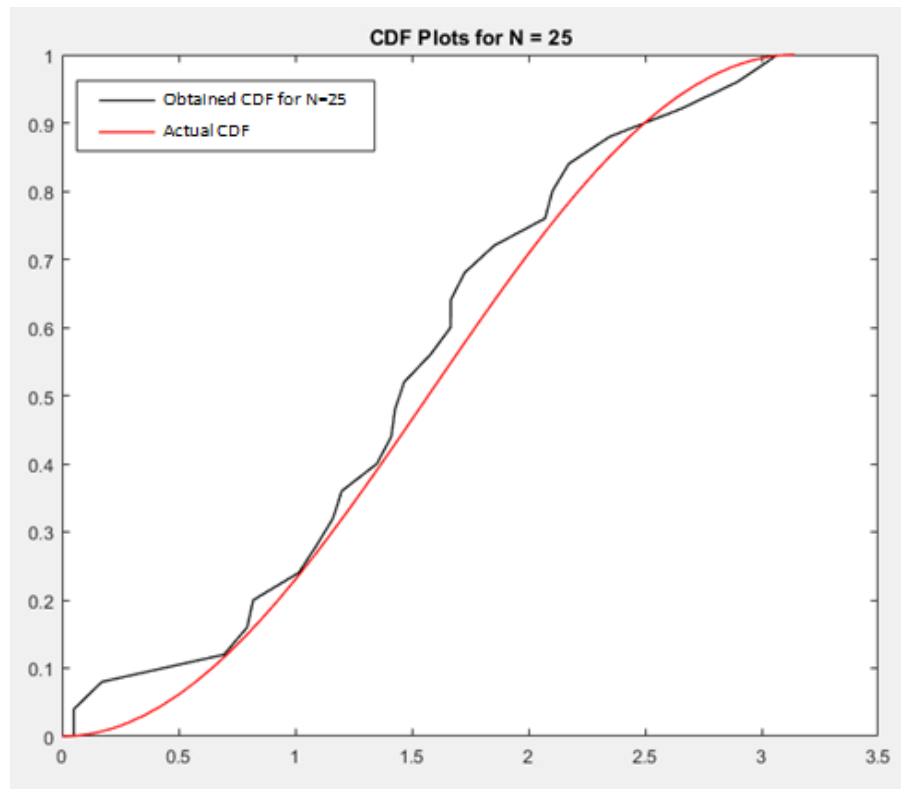


Figure 8 Comparison of CDF Plots for N=25

This process is repeated for N=100 and N=1000. The respective comparison plots are shown in Figure 9 and Figure 10 respectively.

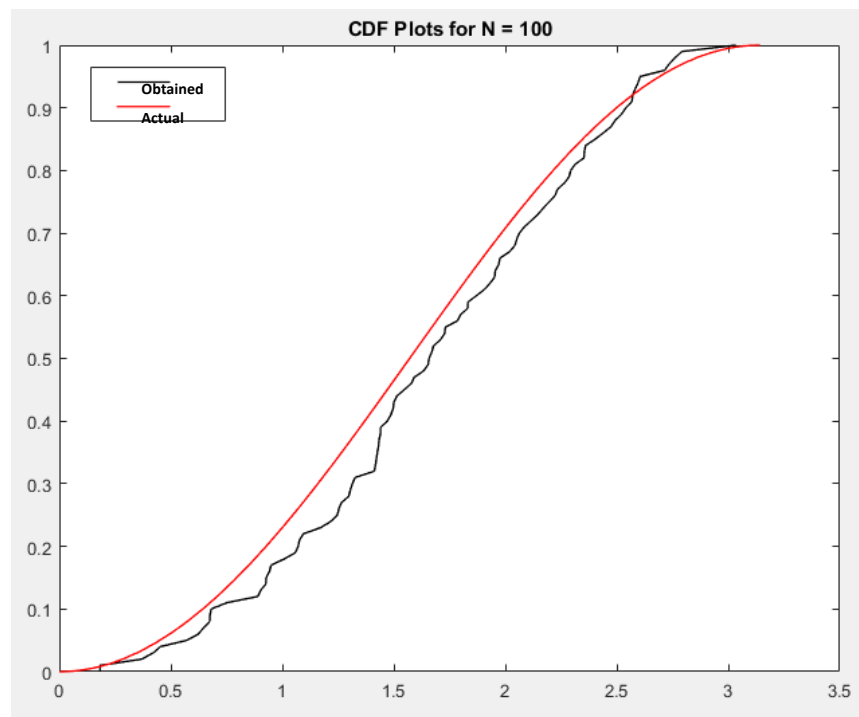


Figure 9 Comparison of CDF Plots for N=100

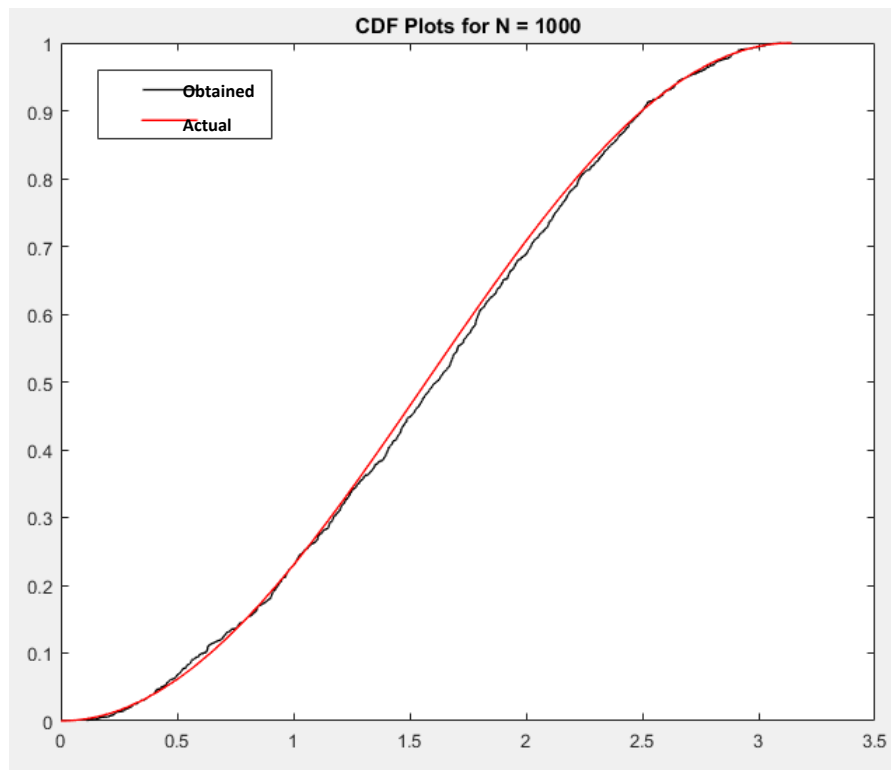


Figure 10 Comparison of CDF Plots for N=1000

For the case N=100 samples, the sample mean and population variance is computed. MSE is computed in two ways – using the population variance and using the sample variance.

First MSE is computed using the population variance. Next assuming the population variance is unknown, MSE of the mean is estimated using sample variance. This is shown in Figure 11.

```
Command Window

s_mean =

    1.6758

MSE using Population Variance :

mse_p_var =

    0.0042

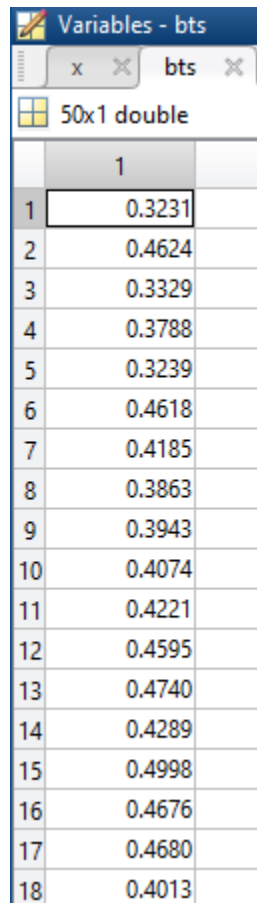
MSE using Sample Variance :

mse_s_var =

    0.0043
```

Figure 11 MSE using Population variance and sample variance.

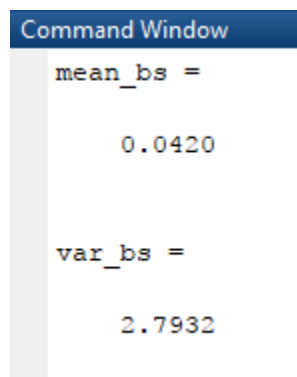
For the case $N=100$, $M=50$ bootstrap samples are generated using the bootstrap command. Figure 12 shows a screenshot of the generated bootstrap samples.



	1
1	0.3231
2	0.4624
3	0.3329
4	0.3788
5	0.3239
6	0.4618
7	0.4185
8	0.3863
9	0.3943
10	0.4074
11	0.4221
12	0.4595
13	0.4740
14	0.4289
15	0.4998
16	0.4676
17	0.4680
18	0.4013

Figure 12 Generated Bootstrap Samples($M=50$) for $N=100$

The sample mean and sample variance is calculated, as shown in Figure 13.



```
Command Window
mean_bs =
    0.0420

var_bs =
    2.7932
```

Figure 13 Computed Sample Mean and Variance for $M=50$

The bootstrapping method is also tried for some different number of samples. Figure 14(a), Figure 14(b) and Figure 14(c) show the mean and variance computed for $M=100$, $M=1000$ and $M=10000$ samples respectively.

```

Command Window
act_mean =

    1.9854

act_var =

    0.2451

mean_bs =

    2.0425

var_bs =

    0.3037

mse_bs =

    0.2665

fx >> |

```

Figure 14(a) Mean and variance for M=100

```

Command Window
act_mean =

    1.5813

act_var =

    0.4285

mean_bs =

    1.2457

var_bs =

    0.3245

mse_bs =

    0.5864

fx >> |

```

Figure 14(b) Mean and variance for M=1000

```

Command Window
act_mean =

    1.3287

act_var =

    0.1242

mean_bs =

    1.4921

var_bs =

    0.1140

mse_bs =

    0.1206

fx >> |

```

Figure 14(c) Mean and variance for M=10000

Then using the obtained mean and variance value, MSE for the generated bootstrapped samples is shown in Figure 15. For N=100, N=1000 and N=10000, the MSE values are shown in Figure 14(a), 14(b) and 14(c) respectively.

```

Command Window
>> mse_bs

mse_bs =

    0.0514

fx >> |

```

Figure 15 MSE of the generated bootstrap samples for N=50.

1.4.3 Results

- i. For the given pdf, the value of the normalizing constant is found by integrating the pdf, as shown in Figure 1. The value of $c = 0.5$ is obtained. The pdf equation becomes $0.5 * \sin(x)$. CDF is obtained by integrating the pdf (as in Figure 2), as $0.5 - 0.5 * \cos(x)$. A sample of generated CDF values is shown in Figure 3. The values are non – decreasing, and are in the range $[0,1]$. Hence the distribution is valid.
- ii. In the first part, population mean and population variance are computed using the respective formulae and compared with the values obtained analytically (using the inbuilt commands). This is shown in Figure 4 and Figure 5. The values obtained are shown in Table 1.

Table 1 Comparison of Mean and Variance

	Calculation by	
	Formulae	inbuilt commands
Mean	1.5708	1.5698
Variance	0.4674	0.4621

From Table 1, it is observed that the computed value and the actual value are approximately the same. To be precise,

There is approximately $\frac{\text{modulus}(1.5698-1.5708)}{1.5698} = 0.063\%$ deviation in the mean value.

There is approximately $\frac{\text{modulus}(0.4674-0.4621)}{0.4621} = 0.011\%$ deviation in the value of variance.

- iii. The plot of PDF is the actual function plotted in the interval $[0, \pi]$. CDF can be computed from the PDF by integrating the PDF over the given interval. The CDF should be a non – decreasing function. This is obtained as expected as in Figure 6(b).
- iv. To generate samples of X, the Cumulative Distribution function is inverted. The inverse CDF is shown in Figure 7. Random numbers are chosen. To check its validity, these generated values are again substituted in the CDF and plotted. Figure 8, Figure 9 and Figure 10 show the CDF plots generated for 25 samples, 100 samples and 1000 samples respectively. In Figure 8, for 25 samples, the cdf plot is irregular. The shape of the plot hardly resembles the actual CDF plot. As the number of samples is increased to 100 (as in Figure 9), the CDF plot becomes more regular. This is now almost the desired plot. When the number of samples is further increased (as in Figure 10), the generated plot almost equals the actual plot.
- v. Mean Square Error(MSE) is a parameter to measure the accuracy of the estimated values. MSE depends on the underlying distribution. Hence, the knowledge of some parameters of

the underlying distribution is necessary. In the first case, MSE of the mean is computed from the population variance. In the second case, assuming population variance is unknown, MSE is calculated from the sample variance. Figure 11 compares both.

MSE of mean using population variance	0.0042
MSE of mean using sample variance	0.0043

It is observed that the MSE values are almost the same in the two cases. This is because the sample size is smaller i.e. 100 samples.

- vi. Bootstrapping technique is done to obtain the subset of a sample space. This is done by taking samples from the empirical distribution. Here the distribution of 100 samples is taken, and bootstrap values are generated. Figure 12 shows a sample of generated bootstrap values.

The mean and variance of the samples generated is compared with the actual values in Table 2 for different bootstrap samples (as in Figure 13 and 14).

Table 2 Comparison of mean and variance for different bootstrapped values

	No. of bootstrap samples			
	M=50	M=100	M=1000	M=10000
MSE Values obtained	0.0514	0.2665	0.5864	0.1206

The values are different because of the use of randomly generated values. But the MSE of the samples is expected to decrease as the number of samples become very large.

- vii. Plotting a histogram for the generated bootstrap samples, it is observed that as the number of samples increase, the pdf becomes closer to the actual pdf. The mean of the samples generated is closer to the actual mean for a large number of values i.e. As the number of samples increase, there are more chances of getting mean closer to the actual mean 1.5708. This is shown in Figure 16.

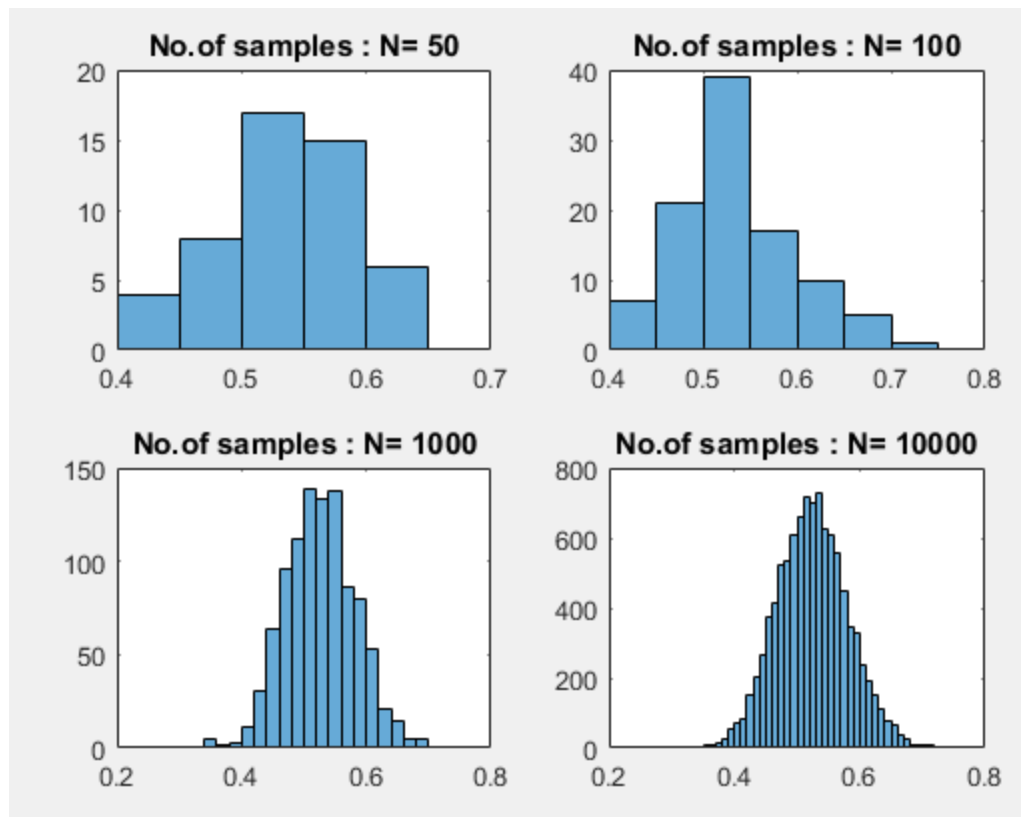


Figure 16 Comparison of the generated bootstrap samples

Thus, statistical analysis of the given continuous distribution is done.

1.5 References

1. Sheldon Ross – Simulation
2. Matlab Tutorials

1.6 Matlab Code

```
clc;clear all;close all;
%define the range
%Find the value of normalizing constant
f1 = @(x) sin(x);
integ1=integral(f1,0,pi);
syms C;
disp('Normalizing constant :');
c=solve(C*integ1==1,C)
%pdf gives the eqn of the pdf function
pdf_func=c*f1

t=0:0.01:pi;
pdf_val=zeros(1,length(t));
for i=1:1:length(t)
```

```

    pdf_val(i)=c*sin(t(i));
end
pdf_val

bins=0:0.1:1;
for i=1:1:length(bins)
    count=0;
    for j=1:1:length(pdf_val)
        if pdf_val(j)<=bins(i)
            count=count+1;
        end
    end
    cdf_val(i)=count/length(pdf_val);
end

cdf_val
plot(cdf_val)

%Calculation of mean and variance
%Calculation of E[X]
f2 = @(x) (0.5*x.*sin(x));
mean_val=integral(f2,0,pi)

%Calculation of E[X^2]
f3 = @(x) (0.5*x.*x.*sin(x));
e_x2=integral(f3,0,pi)

%Calculation of Variance
variance_x=e_x2-(mean_val^2)

%Plot PDF
fplot(pdf_func,[0 pi])

%Finding the Distribution Function
syms x;
cdf_func=int(c*sin (x),0,x);
disp('CDF :');
disp(cdf_func)
disp('Inverse of CDF: ');
inv_func = finverse(cdf_func)
figure
fplot(cdf_func,[0 pi])
N=[25 100 1000];

for i=1:1:length(N)
    figure
    x1=0;y=zeros(1,N(i));
    for j=1:1:N(i)
        x1=rand(1,1);
        syms x;
        y(j)=subs(inv_func,x,x1);
    end
    [f2,x2]=ecdf(y);
    if N(i)==100
        cdf_f_100=f2;
        cdf_f_x=x2;
    end
end

```

```

        cdf_f_y=y;
    end
    plot(x2,f2,'k');
    hold on;
    fplot(cdf_func,[0 pi]);
    hold off
    title(['CDF Plots for N = ',num2str(N(i))]);
end

%Finding the sample mean for N=100
s_mean=sum(cdf_f_y)/length(cdf_f_y)
%Finding MSE using Population variance
%mse=population variance /no.of samples
p_var=var(cdf_f_y);
disp('MSE using Population Variance :');
mse_p_var=p_var/100

%Computing sample variance
v=zeros(1,length(cdf_f_y));
for i=1:length(cdf_f_y)
    v(i)=(cdf_f_y(i)-mean(cdf_f_y))^2;
end
s_var=sum(v)/(100-1);
disp('MSE using Sample Variance :')
mse_s_var=s_var/(100-1)
figure
%Bootstrapping
sam=[50 100 1000 10000];
for k=1:length(sam)
    bts=bootstrp(sam(k),@var,cdf_f_y);

    var_bs1=zeros(1,100);

    act_mean=mean(cdf_f_y)
    act_var=var(cdf_f_y)
    mean_bs=sum(bts)/100
    for i=1:100
        var_bs1(i)=(cdf_f_y(i)-mean_bs)^2;
    end
    var_bs=sum(var_bs1)/100
    % edg=0:0.05:pi;
    subplot(2,2,k)
    histogram(bts);%,length(edg))
    % hold on;
    % fplot(pdf_func,[0 pi],'k');
    % fv=0.5*sin(edg);
    % plot(edg,fv,'k');
    % hold off;
    title(['No.of samples : N= ',num2str(sam(k))]);
    mse_bs=var_bs/(sam(k))
end

```