EE 511 – Project #1 Muthulakshmi Chandrasekaran

September 20, 2017 USC ID 4486180242

1.1 Problem Statement

To generate a sequence of random numbers in the range [0,1] in Matlab using the $rand$ function.

Part A: To calculate sample mean and sample variance for the generated sequence, and to compare the results with the expected value.

Part B: To calculate sample covariance, and to test the independence of the generated sequence, using the behavior of covariance.

Part C: To use the data to generate a discrete random variable that is uniformly distributed in [0,9], and to analyze the frequency distribution plots.

1.2 Mathematical Basis

A Random Variable is a mapping from the set of outcomes to the set of numbers. The mapping could be discrete or continuous. In Matlab, random numbers are generated using the rand function. MATLAB uses algorithms that make the generation of random numbers appear to be random and independent. These apparently random and independent numbers are often described as pseudorandom and pseudo – Independent.

A Uniform Random Variable has each outcome equally likely, and values are uniformly distributed throughout some interval. Consider a uniformly distributed random variable distributed as $\{ x_1, x_2, x_3, \ldots, x_M\}$ on the interval $[a, b]$. The probability of each outcome is given as

$$Probablity = \frac{1}{M}$$

The mean of a Uniform distribution in the range $[a, b]$ is given as

$$Mean = \frac{a+b}{2}$$

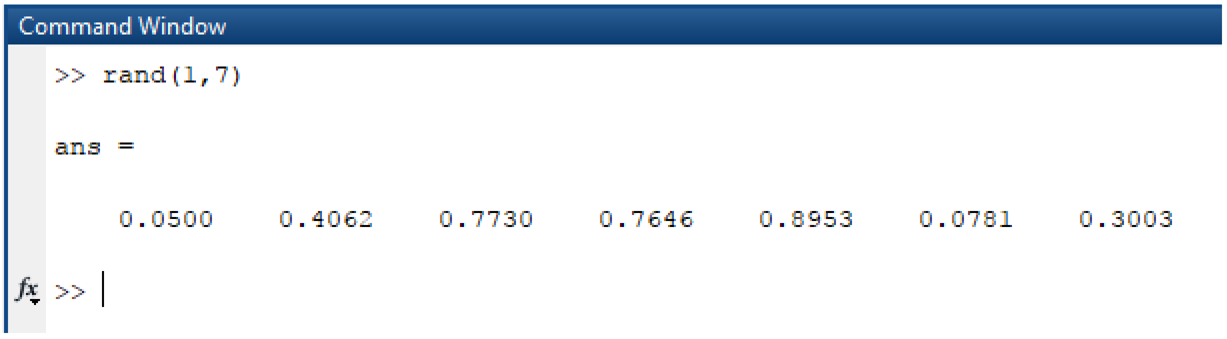
Variance of the uniform random variable distributed in the range $[a, b]$ is given as

$$Variance = \frac{(b-a)^2}{12}$$

1.3 Simulation in Matlab

1.3.1 Generation of a sequence of random numbers in Matlab

In Matlab, random numbers are generated using the $rand$ command. The function $rand$ generates uniformly distributed random numbers. The function $rand(M,N)$ generates a MxN matrix of uniformly distributed random numbers. An example of random number generation is shown in Figure 1.



```
>> rand(1,7)

ans =

    0.0500    0.4062    0.7730    0.7646    0.8953    0.0781    0.3003

>> 
```

Figure 1 Generation of random numbers in Matlab using the function $rand$

1.3.2 Calculation of Mean and Variance – Part A

Sample mean and sample variance is calculated for the generated sequence, as in Program 1 in Section 1.6. It is then compared with the expected value.

Description of the code:

A sequence of N random numbers, (here N=1000000), is generated using the $rand$ function. Sample mean $\bar{x}$ of a sequence of M samples, say x_1, x_2, x_3,…., x_M is given by

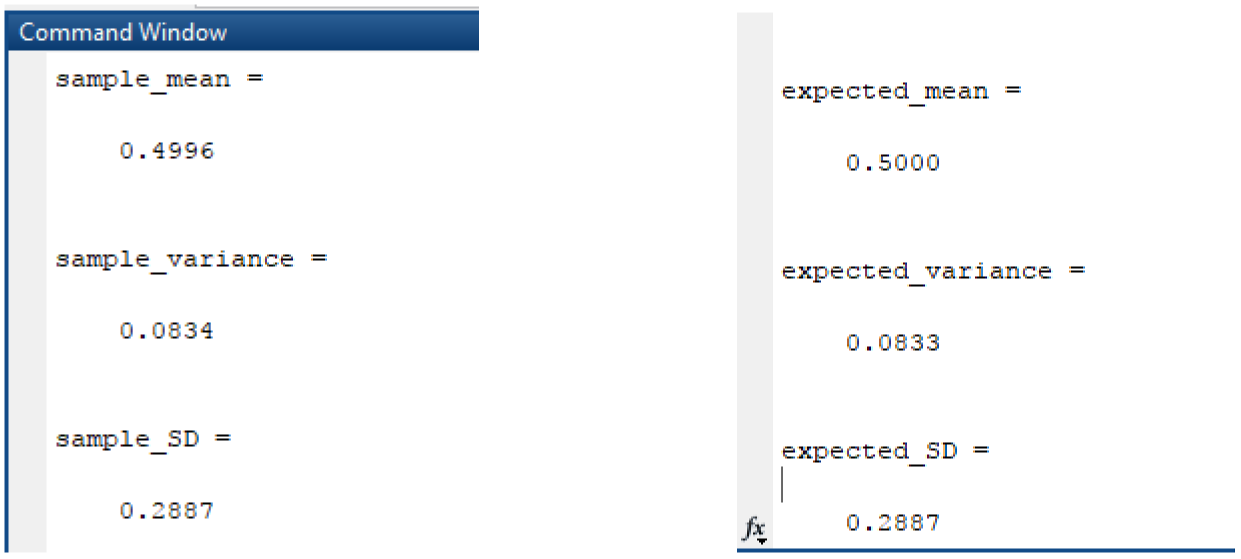
$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_M}{M}$$

Sample Variance σ^2 is calculated as

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M}(x_i - \bar{x})$$

The sample mean of the sequence is calculated using the built – in $mean$ function. Sample variance is mathematically calculated. The expected mean value and the expected Variance are calculated from the standard formulae for Uniform Distribution.

Figure 2 shows a sample of the output obtained for one trial.



```
sample_mean =

    0.4996


sample_variance =

    0.0834


sample_SD =

    0.2887
```

```
expected_mean =

    0.5000


expected_variance =

    0.0833


expected_SD =

    0.2887
```

Figure 2 Sample Output for Observed and Expected Values of Mean and Variance

Results:

Comparing the observed and expected values of mean and variance, it is observed that both are almost equal. The upper bound of N depends on the memory availability. Figure 3 shows the histogram plot of the values obtained. It is observed that the values generated are distributed in a uniform fashion over the interval [0,1]. This shows that the outcomes are equally likely, and thus following a uniform distribution.

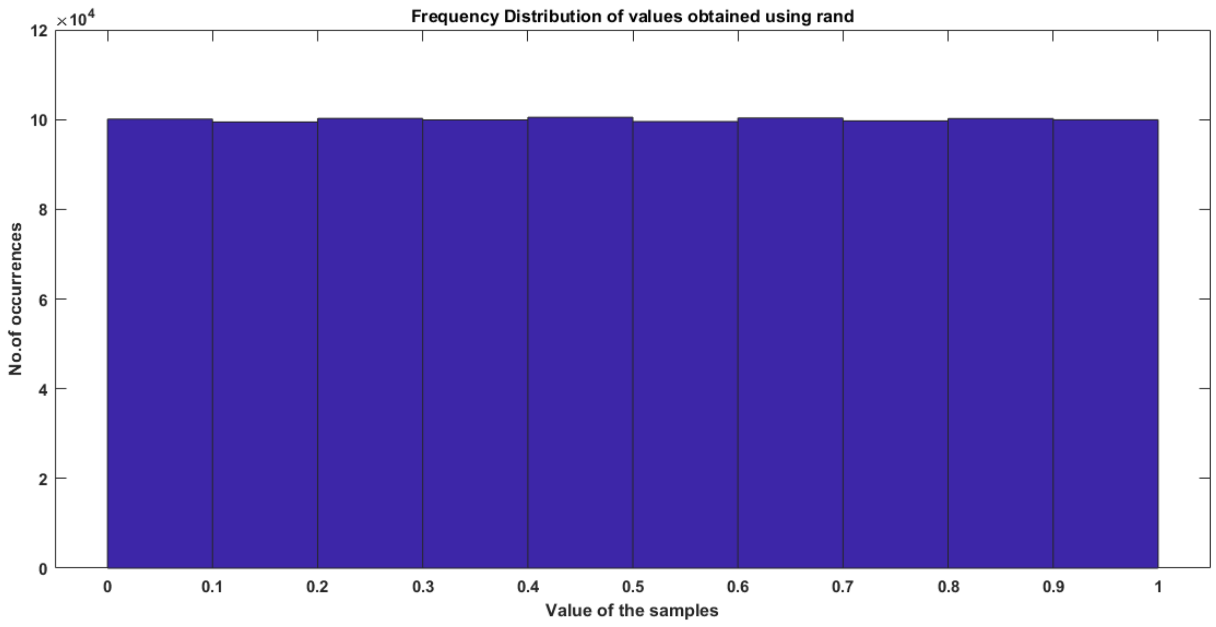


Figure 3 Frequency Distribution Plot of the sequence obtained

1.3.3 Calculation of Covariance – Part B

Covariance is a measure of how two variables are related. From the covariance parameter, a notion of independence is observed. A positive covariance means the variables are positively related, while a negative covariance means the variables are inversely related.

Covariance between two random variables x and y , of total samples N is calculated as

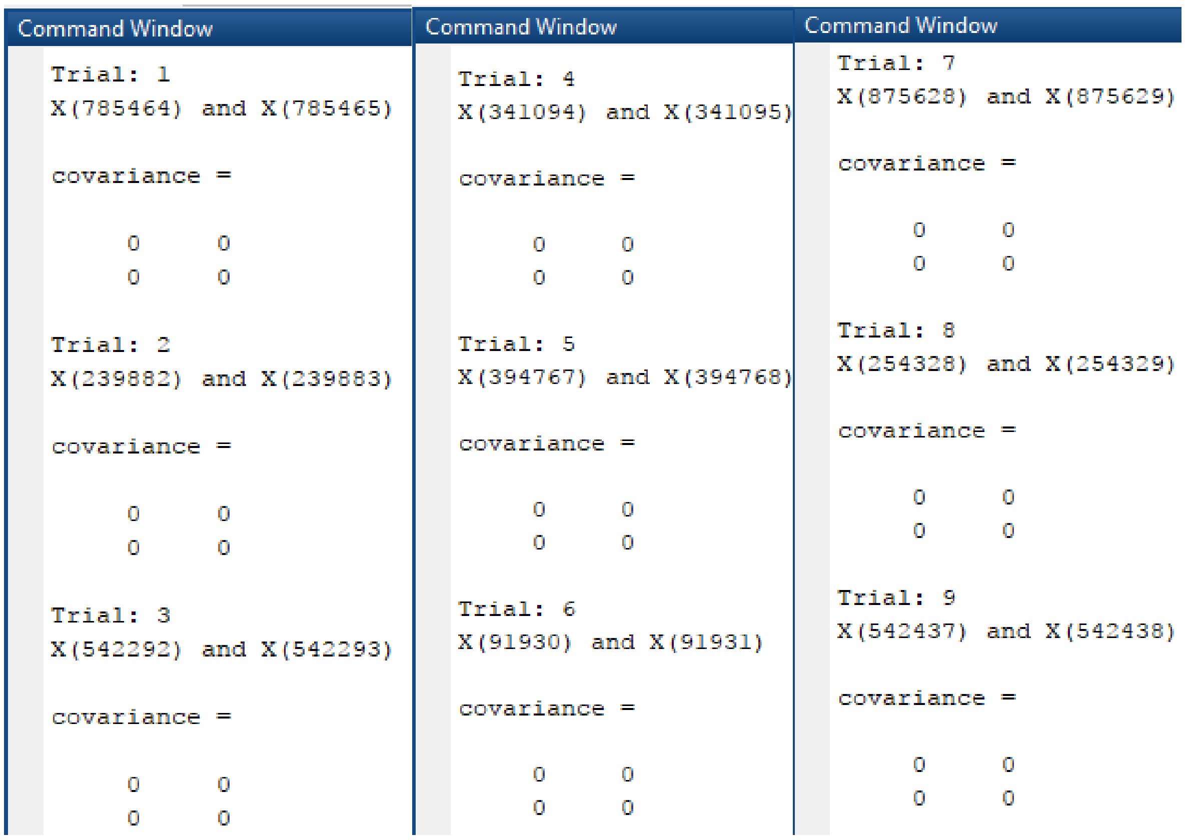
$$Cov(x, y) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})\,(y_i - \bar{y})$$

where $\bar{x}$ and $\bar{y}$ are the mean of the random variables x and y, respectively.

<u>Description of the code:</u>

A sequence of random numbers is generated using the $randn$ function. Since $randn$ is used to generate uniformly distributed values in the range [0,1], each value has a probability of $\frac{1}{N}$. Covariance measures the relationship between two variables. In general, $Cov(X, X) = Var(X)$.

Here, to find $Cov(X_k, X_{k+1})$, some random values for k are chosen using the $randi$ command. The function $randi(max)$ choses a random integer value between 1 and max. This process is iterated for 9 times, each time $randi$ yielding a different value of k. Covariance is found using the built – in $cov(A, B)$ command, as in program 2 in Section 1.6. Figure 4 gives the command window screenshots of the observed trials.



```
Trial: 1
X(785464) and X(785465)

covariance =

     0     0
     0     0

Trial: 2
X(239882) and X(239883)

covariance =

     0     0
     0     0

Trial: 3
X(542292) and X(542293)

covariance =

     0     0
     0     0
```

```
Trial: 4
X(341094) and X(341095)

covariance =

     0     0
     0     0

Trial: 5
X(394767) and X(394768)

covariance =

     0     0
     0     0

Trial: 6
X(91930) and X(91931)

covariance =

     0     0
     0     0
```

```
Trial: 7
X(875628) and X(875629)

covariance =

     0     0
     0     0

Trial: 8
X(254328) and X(254329)

covariance =

     0     0
     0     0

Trial: 9
X(542437) and X(542438)

covariance =

     0     0
     0     0
```

Figure 4 Covariance observations for 9 trials

Note on Independence:

If X_k and X_{k+1} are independent, the covariance should be equal to zero. This has been shown in the above simulation, where $randn$ is used to generate a set of uniformly distributed random numbers. Since $randn$ generates a sequence where each element is statistically independent of the preceding element, the order of choosing a value of k doesn't matter. Since the elements are statistically independent, the covariance turns out to be zero.

1.3.4 Generation of a Random Variable – Part C

A discrete random variable is to be generated from the sequence generated by the $rand$ function, and compared with the hypothesized distribution. The hypothesized distribution assumes a probability of $\frac{1}{N}$ for each outcome.

<u>Description of the code:</u>

A sequence of random numbers is generated using the $rand$ function. It is then mapped to the set of discrete random variable [0,9] with uniform distribution. The ceil function maps the obtained values into the set of natural numbers. This is shown in Program 3 in Section 1.6. The histogram plot of the generated random variable is shown in Figure 5. The generated plot confirms Uniform distribution, since all values show equal probability of occurrence.

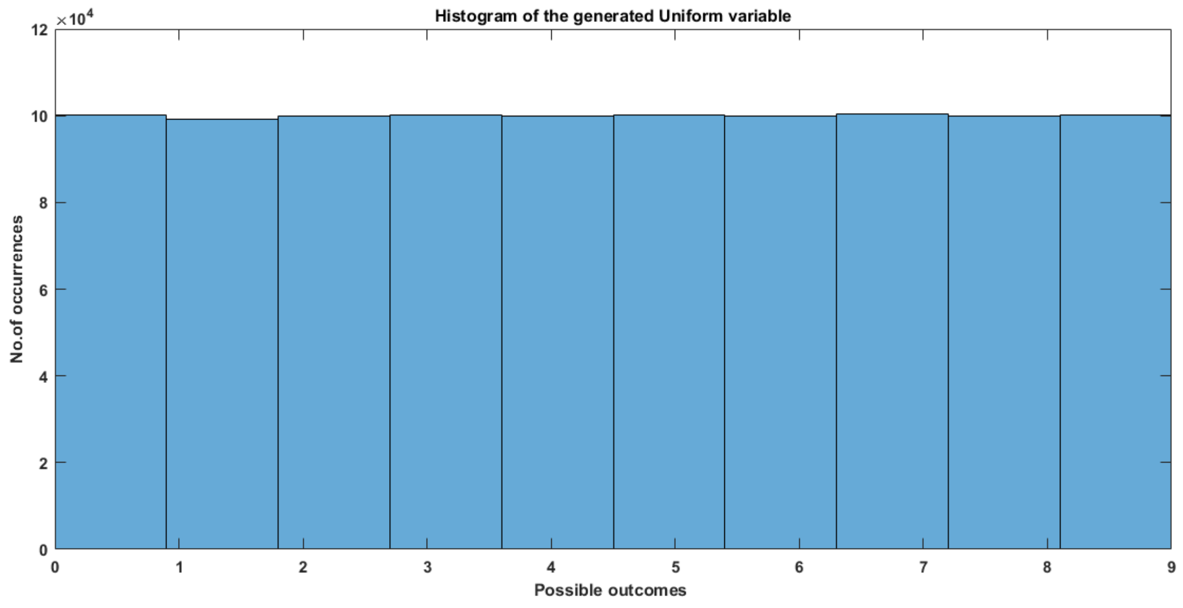


Figure 5 Histogram of the generated Uniform Discrete Variable

Goodness of fit test:

A probabilistic analysis begins with hypothesizing some of the random elements. These hypotheses can be statistically testing the given data using goodness-of-fit tests, and compare how consistent the hypothesis is.

Every hypothesis test requires a null hypothesis (H_0) and an alternative hypothesis (H_a). For a chi-square goodness of fit test, the hypotheses take the following form.

H_0: The data are consistent with a specified distribution.

H_a: The data are not consistent with a specified distribution.

The chi-square test statistics is calculated using the formula:

$$\chi^2 = \frac{(Observed\,Value - Expected\,Value)^2}{Expected\,Value}$$

For one iteration, the values are obtained as follows

Value of the RV	Expected Value E	Obtained Value O	$O - E$	$(O-E)^2$	$\frac{(O-E)^2}{E}$
0	100000	99640	-360	129600	1.296
1	100000	99834	-166	27556	0.27556
2	100000	100122	122	14884	0.14884
3	100000	100208	208	43264	0.43264
4	100000	99936	-64	4096	0.04096
5	100000	99454	-546	298116	2.98116
6	100000	100000	0	0	0
7	100000	100327	327	106929	1.06929
8	100000	100212	212	44944	0.44944
9	100000	100267	267	71289	0.71289
Value of p					7.40678

Figure 6 shows a comparison of the expected and observed outcomes. The value of p = 7.40678 is looked up on the table. Here, there are nine degrees of freedom, since there are 10 outcomes. Looking up the table, it is found that the value of p = 7.40678 takes a value of almost 0.6, which is not significant at a significance level <0.5. The probability of committing an error, when a null hypothesis is rejected even when it is true, is called the significance level. But this value is significant for level>0.5.

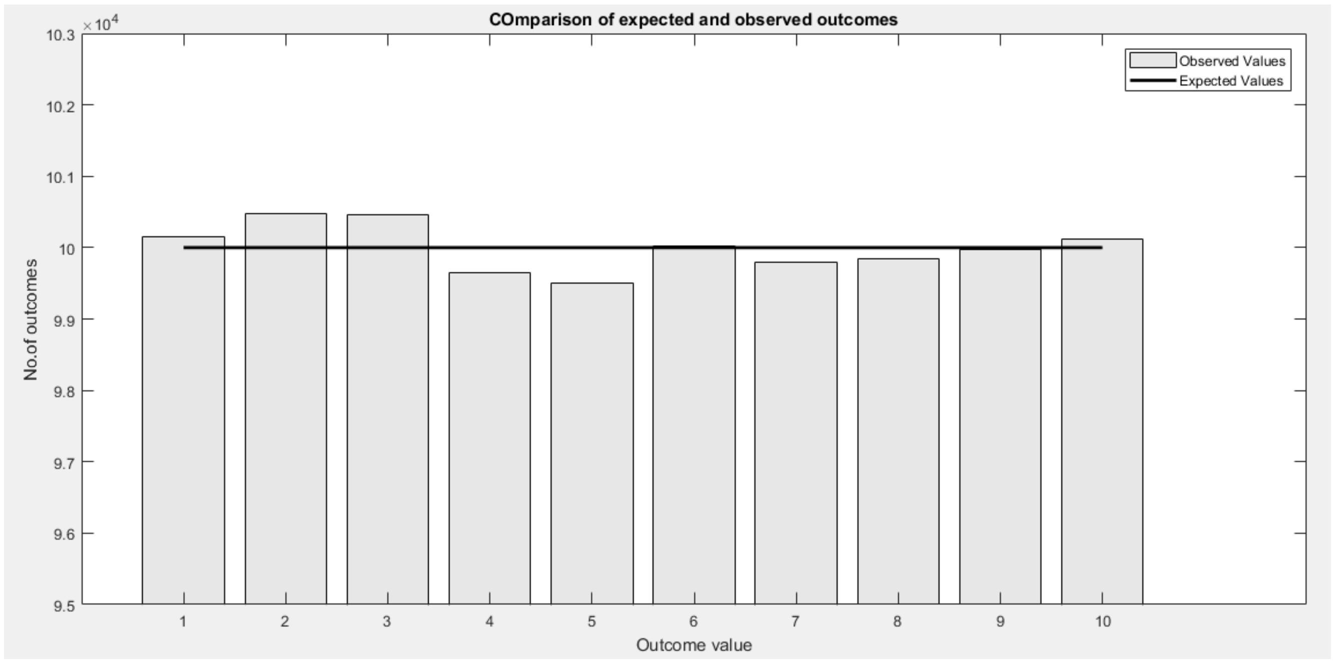


Figure 6 Comparison of the expected and observed outcomes

1.4 Results and Conclusion

1. A random sequence is generated using the rand function. The upper bound of the maximum number of values generated is on the memory availability.
2. Mean and variance of the generated sequence is compared with the expected values of mean and variance, which are found to be almost equal. Since the generated variable is Uniform, each outcome has an equally likely occurrence.
3. Covariance is found. Since the consecutive elements are independent (by the property of random number generation), the covariance turns out to be zero.
4. A discrete random variable is generated from the random sequence. Goodness of fit test is done, to check the consistency of the observed values with the hypothesis. In the trial done, it is observed that this set of outcomes has almost 60% chance of occurrence.

1.5 Reference

1. Sheldon M Ross, "Simulation", 5th edition
2. www.mathworks.com

1.6 Matlab Codes for the Experiment

Program 1 – Calculation of Mean and Variance:

```
% Program to find the Mean and Variance
clc; clear all;close all;

% total number of samples = N
N=1000000;
sample_mean=[];
sample_variance=[];
for k=1:1:20
      A=rand(1,N);

      % Plot the frequency distribution
      hist(A)

      %calculation of sample mean
      sample_mean=[mean(A) sample_mean];
      x=mean(A);

      %calculation of sample variance
      sum=0;
      for i=1:N
          sum=sum+((A(i)-x)^2);
      end
      sample_variance= [sum/(N-1) sample_variance];

      %calculation of sample Standard Deviation
      sample_SD=sqrt(sample_variance)
end

% values obtained using formulae
a=0;b=1;
expected_mean=(1/2)*(a+b)
```

```
expected_variance=(1/12)*((b-a)^2)
expected_SD=sqrt(expected_variance)
```

Program 2 : Calculation of Covariance

```
%Program to calculate covariance
clc; clear all;close all;

% total number of samples=N
N=1000000;
A=rand(1,N);

% Plot the frequency distribution
hist(A)

for i=1:10
    disp(['Trial: ',num2str(i)]);
    k=randi(N);
    x=A(k);
    if k>=N
        k=N;
    end
    x1=A(k+1);
    disp(['X(',num2str(k),') and X(',num2str(k+1),')']);
    % covariance is found using the built in matlab function
     covariance=cov(x,x1)
end
```

Program 3 : Generation of Discrete Uniform Random Variable

```
% Program to generate discrete random variables
clc; clear all;close all;
N=1000000;
A=rand(1,N);
```

```
%Generation of Discrete Random variable
urv=[ceil(-1+(10-0)*A)];
figure

% Plot the frequency distribution
histogram(urv,10)
val=[];

% Goodness of fit test - formula
for i=1:10
   val=[nnz(urv==i-1) val];
end
%Display output
val=val'
```