**PROJECT #2 – SAMPLING & STATISTICS**

**Divyanshu Sahay  
(dsahay@usc.edu)**

Important Note:

Tool Used: **MATLAB R2019a**

1. **Uniform Sampling on an Interval:**

**Problem Statement:**  Uniform Sampling has to be simulated in the interval **[-3,2]**. Following should be the findings:

* Histogram of the Uniform Sampled Outcome.
* Comparison of Sample Mean and Sample Variance with their theoretical values.
* Computation of Bootstrap Confidence Interval for the Sample Mean and Sample Standard Deviation.

**Solution:**

**Understanding Uniform Distribution:** As the name suggests the Uniform Distribution refers to a *family of symmetric probability distribution* *(Ref: Wiki)*. This means, that the probability of occurrence of a members of a sample is equal. Interestingly, uniformity is in fact a natural behaviour which is observed as a result of increase in number of samples.

**Methodology (1.a):** In the first part, it is asked to generate histogram of the sampled outcome.

In MATLAB the random value generator function **rand** can be used to do the same. As the sampling must be done in the specified interval, an equation is formed to make it possible.

**Program Written:**

% The interval [-3,2] can be represented as x and y

x = -3;

y = 2;

Total\_Sample = 50000; % Number of samples taken between the given interval

Sampling = x + (y-x)\*rand(Total\_Sample,1); % Sampling between the interval given by y-x

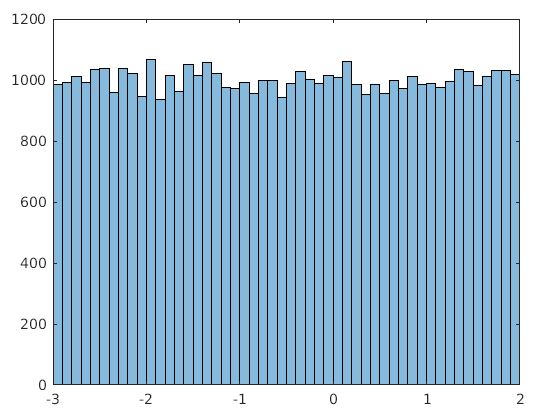
figure(1);

grid on;

histogram(Sampling);

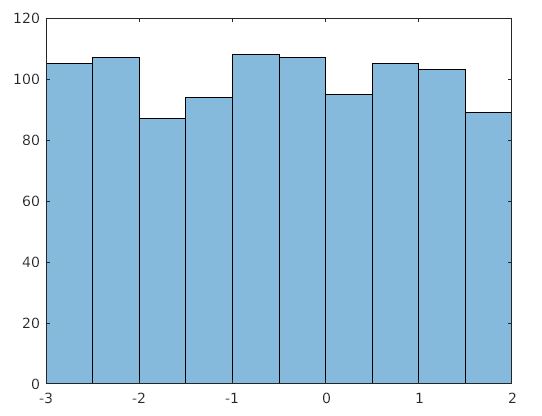
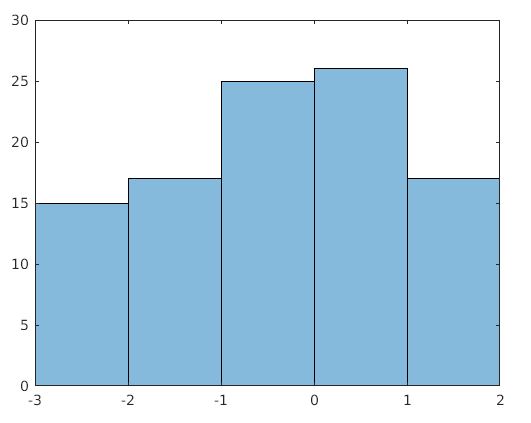
xlim([x y]);

**Results and Observations (1.a):** It can be seen in Fig.1. that the histogram that as probability of occurrence of every sampled data is almost the same. As aforementioned, this is a direct outcome of choosing large number of samples. We can observe in the experiments with less number of samples that the probability of occurrence is fairly varying. Thus, the concept of Uniform Distribution is best understood with sample size being 50,000

****Fig.1. Uniform Sampling observed for Total\_Sample = 50,000

Value of Member

Frequency of Sample Member

** **  
Fig. 2.Non-Uniform Distributuon observed for Total\_Sample being 1000 (left) and 100(right)

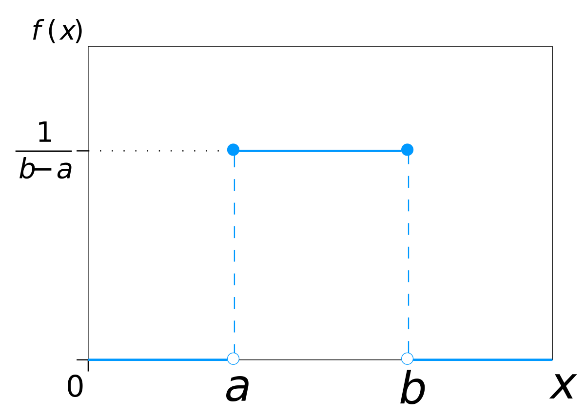
Frequency of Sample Member

Value of Member

Frequency of Sample Member

Value of Member

**Mean and Variance of Uniformly Distributed Function:** It is evident that a uniform probability distribution between an interval ***a*** and ***b*** can be represented as shown in Fig. 3.

****  Fig.3. Uniform Probability Distribution

The Probability Density function can be written as,

Eq.(1)

We know that, the Mean (or Expectation) can be obtained as follows

Eq.(2)

When we substitute Eq. (1) in Eq. (2), we get the Mean as,

Eq.(3)

Subsequently, the Variance can be obtained by,

Eq.(4)

Thus, Eq. 3 and Eq. 4 can be used to calculate the Mean and Variance.

**Methodology (1.b):** For this the part the equations derived above can be used directly in MATAB to calculate the theoretical values. T0he Sample Mean and Sample Variance can be calculates using the MATLAB command ***mean()***and ***var()*.** The code was run repeatedly for different sample values to obtain respective outputs.

**Program Written:**

% % The interval [-3,2] can be represented as x and y

x = -3;

y = 2;

Total\_Sample = 1000; % Number of samples taken between the given interval

Sampling = x + (y-x)\*rand(Total\_Sample,1); % Sampling between the interval given by y-x

% Theoretical Mean

Mean\_th = (x+y)/2;

% Theoretical Variance and standard deviation

Var\_th = (y-x)^2/12;

SD\_th = sqrt(Var\_th);

% Sample parameters are calculated Using MATLAB Command

% SAMPLE MEAN:

Mean\_Sample = mean(Sampling);

% SAMPLE VARIANCE:

Var\_Sample = var(Sampling);

% STANDARD DEVIATION

SD\_Sample = std(Sampling);

% Results

disp('THEORETICAL VALES')

fprintf('Theoretical Mean: %f \n', Mean\_th);

fprintf('Theoretical Variance: %f \n',Var\_th );

fprintf('Theoretical Standard Deviation: %f \n', SD\_th);

fprintf('\n');

fprintf('\n');

disp('SAMPLE VALES')

disp('---1000 Samples-----')

fprintf('Sample Mean: %f \n', Mean\_Sample);

fprintf('Sample Variance: %f \n', Var\_Sample);

fprintf('Sample Standard Deviation: %f \n', SD\_Sample);

**Results and Observations (1.b):**

**Command Window:**

THEORETICAL VALUES

Theoretical Mean: -0.500000   
Theoretical Variance: 2.083333   
Theoretical Standard Deviation: 1.443376

SAMPLE VALUES  
--- 100 Samples-----  
Sample Mean: -0.478960   
Sample Variance: 2.318049   
Sample Standard Deviation: 1.522514

--- 1000 Samples-----  
Sample Mean: -0.520103   
Sample Variance: 2.016977   
Sample Standard Deviation: 1.420203

--- 50000 Samples-----  
Sample Mean: -0.497094   
Sample Variance: 2.087947   
Sample Standard Deviation: 1.444973

A comparison between the values of the samples and theoretically obtained values reflect that, the values with largest number of samples are closest to theoretical values. This is in congruence with our earlier argument that the uniformity is nothing but a result of taking many samples, thereby creating a Uniform Probability Distribution.

**Understanding Bootstrap Confidence Interval:** There are two terms here which needs justice in terms of explanation, *Bootstrap* and *Confidence Interval*. Turns out, both have an exceptionally important role to play in Probability and Statistics.

Bootstrap: Scary it may sound, simply refers to Sampling with Replacement. By replacement, it means that the Same Number is allowed to be sampled again in a sample interval. As wiki says, *The basic idea of bootstrapping is that inference about a population from sample data  
(sample → population) can be modelled by resampling the sample data and performing inference about a sample from resampled data (resampled → sample)*. It helps us measure the reliability of our statistic. It will get more clear as we analyse the output for task 1(c).

Confidence Interval: As the name suggest, it is an *Interval* of data that may or may not contain an unknown population parameter of interest. There is certain level of *Confidence* that is associated with that assertion when the experiment is repeated with different intervals for the same population parameter. Typically, a confidence interval is provided with a confidence level of 95%. A Confidence level of 95% also suggest that a significance level of 5% has been considered.

**Methodology (1.c):** The boot strap confidence interval for the sample mean and standard deviation can be calculated using the MATLAB command "***bootci***", which considers the 95% bootstrap confidence interval. The methodology for this part is explained mostly in the comments given in the program.

**Program Written:**

%BOOTSTRAP INTERVAL FOR THE SAMPLE MEAN.

[CI\_Mean, MEAN\_CI] = bootci(10000,@mean,Sampling);

% CI\_Mean (user\_specified\_name) is the parameter where the bootci returns the confidence intervals

% MEAN\_CI (user\_specified\_name) is the parameter which receives the Sample mean value

% in bootci: 10,000 is the user\_specified number of samples

% and 'Sampling' is the randomly generated samples between x and y.

% @mean is the fuction handle that determines the operation that will be

% performed by the bootci function.

disp('CONFIDENCE INTERVALS FOR SAMPLE MEAN')

fprintf('2.5th : %f \n', CI\_Mean(1));

fprintf('97.5th : %f \n', CI\_Mean(2));

figure(1);

histogram(MEAN\_CI);

hold on; % lets us plot both the confidence interval and histogram on the same plot

plot((mean(MEAN\_CI))\*[1,1],ylim,'LineWidth',1.5);

plot((CI\_Mean(1))\*[1,1],ylim,'LineWidth',1.5);

plot((CI\_Mean(2))\*[1,1],ylim,'LineWidth', 1.5);

%BOOTSTRAP INTERVAL FOR THE STANDARD DEVIATION.

[CI\_SD, StdD\_CI] = bootci(10000,@std,Sampling);

% CI\_SD (user\_specified\_name) is the parameter where the bootci returns the confidence intervals

% StD\_CI (user\_specified\_name) is the parameter which receives the Sample Standard Deviation value

% in bootci: 10,000 is the user\_specified number of samples

% and 'Sampling' is the randomly generated samples between x and y.

% @std is the fuction handle that determines the operation that will be

% performed by the bootci function.

disp('CONFIDENCE INTERVALS FOR SAMPLE STANDARD DEVIATION')

fprintf('2.5th : %f \n', CI\_SD(1));

fprintf('97.5th : %f \n', CI\_SD(2));

figure(2);

histogram(StdD\_CI);

hold on; % lets us plot both the confidence interval and histogram on the same plot

plot((CI\_SD(1))\*[1,1], ylim,'LineWidth',1.5);

plot((CI\_SD(2))\*[1,1], ylim,'LineWidth',1.5);

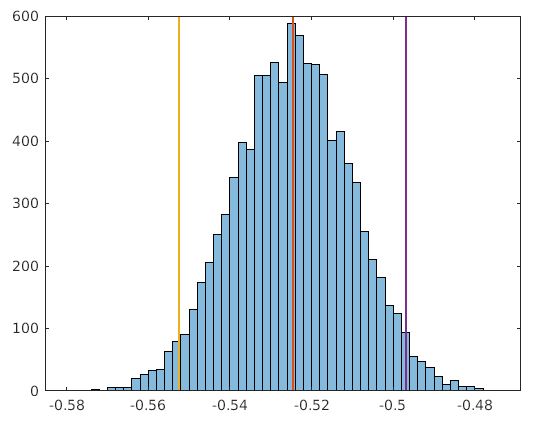
**Results and Observations (1.c):**

**Command Window:**

CONFIDENCE INTERVALS FOR SAMPLE MEAN  
2.5th : -0.532773   
97.5th : -0.476723

CONFIDENCE INTERVALS FOR SAMPLE STANDARD DEVIATION  
2.5th : 1.435976   
97.5th : 1.461193

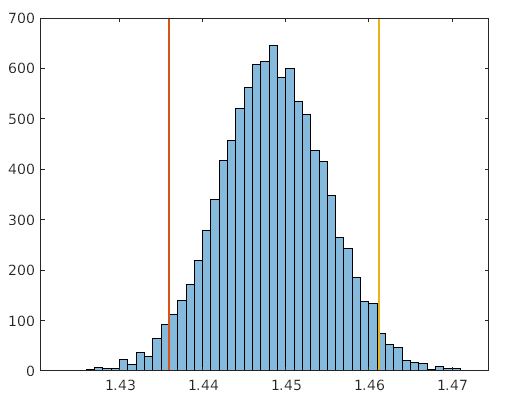
The 95% confidence is composed by the two extreme points of 97.5th percentile and 2.5th percentile (thereby giving 97.5 – 2.5 = 95%). Same can be observed in the plots below.

  
Fig. 4. Mean of Samples with 95% CI

Mean of Samples

Number of Samples

**95% CI**

  
Fig. 5. Standard Deviation of Samples with 95% CI

Standard Deviation of Samples

Number of Samples

**95% CI**

The plots provide a good estimate of the population mean and standard deviation, the quantities which are generally unknown. Bootstrapping as stated earlier strengthens the reliability of statistics.

1. **Covariance and Correlation:**

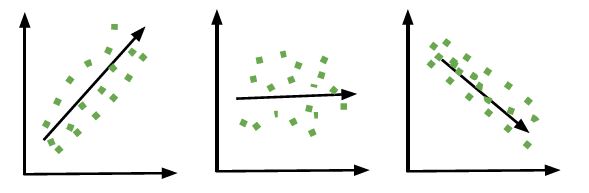
**Problem Statement:**  A sequence ***X*** has to be generated by drawing sample from a uniform random variable. The problem asks to perform following operations:

* The Covariance for and , where is right shifted sequence of .
* A new sequence has to be generated and its covariance with has to be determined.

**Solution:**

**Understanding Covariance and Correlation:** Both the parameters are used to describe a certain kind of relationship between two random variables. They tell about the degree to which two random variables deviate from an expected mean value. Let’s examine an image to understand it better and further quantify the understanding with some mathematical equations.

**y**

Fig. 6. Understanding Covariance and Correlation *(Courtesy: Google Images).*

**(iii)**

**(ii)**

**(i)**

**X**

In Fig. 6, we can see that there are three graphs which indicates the relationship between the covariance/correlation between X and Y. The first image (i) depicts the scenario of strictly positive covariance/correlation between the random variables X and Y. This means, X and Y has similar behaviour. In other words, If X is greater than its mean value Y will also be greater than it’s mean. Similar explanation can be made for (ii) and (iii) which depicts Zero and Strictly Negative Covariance/Correlation between X and Y respectively.

The above explanation might give a false interpretation that Covariance and Correlation are same. However, there is a significant difference between the two. Following mathematical explanation will make it more clear.

Covariance is given by:

It can be observed in the above equation that Covariance will have units which will be dependent on X and Y. This can lead to a possible discrepancy when these random variables are of completely different types and the combined unit will not make much sense. Therefor the notion of Correlation is introduced which is a unit less quantity obtained by normalizing the above equation. So the Correlation is given as,

Now, it can be seen that this normalized quantity will be unit less and can help understand the relationship between ***X*** and ***Y* in** a more sensible way. Also, since its normalized it can be stated that the value of will be between -1 and 1, i.e. [-1,1]. Where a value of -1 will suggest a strictly negative correlation, 1 will suggest a positive correlation and 0 will suggest independence.

**Methodology:**  The major requirement of this problem was performing left and right shifts to the random variable set ***X.*** The program written below will explain the approach adopted. Both the requirements of 3 a. and 3 b. is addressed by the program.

**Program Written:**

samples = 1000;

%random vaiable

Xk = rand(1, samples);

%random variable shifted right by 1

Xk\_plus\_1 = zeros(size(Xk));

shift = 1;

Xk\_plus\_1(shift+1:end)=Xk(1:end-shift);

% calculationn of covariance

Covariance\_2a = cov(Xk,Xk\_plus\_1);

disp(Covariance\_2a);

%random variable shifted left by 1

Xk\_minus\_1 = zeros(size(Xk));

Xk\_minus\_1(1:end-shift)=Xk(2:end);

%random variable shifted left by 2

shift\_2 = 2;

Xk\_minus\_2 = zeros(size(Xk));

Xk\_minus\_2(1:end-shift\_2)=Xk(3:end);

%random variable shifted left by 3

shift\_3 = 3;

Xk\_minus\_3 = zeros(size(Xk));

Xk\_minus\_3(1:end-shift\_3)=Xk(4:end);

%performing substraction

Yk = Xk - 2\*Xk\_minus\_1 + 0.5\*Xk\_minus\_2 - Xk\_minus\_3;

Covariance\_2b = cov(Xk, Yk);

disp(Covariance\_2b);

**Results and Observations:**

**Command Window:**

Covariance of Xk and X(K+1)  
 0.0839 0.0003  
 0.0003 0.0839

Covariance of Xk and Yk  
 0.0839 0.0833  
 0.0833 0.5235

MATLAB chooses to displays the Covariance as follows

**Cov(X,X) Cov(X,Y)**

**Cov(Y,X) Cov(Y,Y)**

So, it can be observed in the case ofand that the covariance is very close to zero.  
Thus, they can be considered uncorrelated. Whereas, in the case of and the value of covariance is small clearly not very close to zero. Therefore, and are correlated.

1. **Goodness-of-fit Test:**

**Problem Statement:**  It is asked to simulate sampling of the outcome 0,1,2 … M-1 with replacement. Following are the tasks assigned:

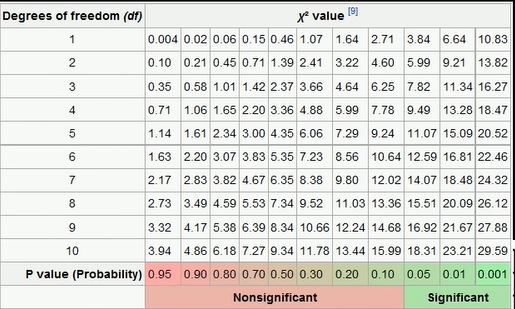
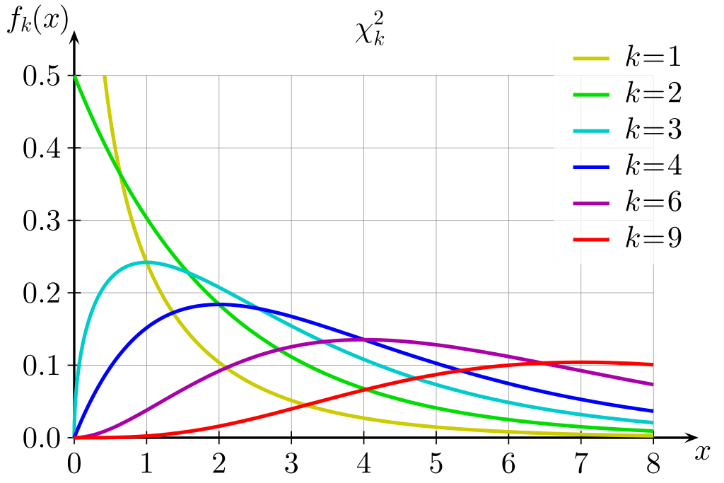
* Histogram of the Outcomes.
* Statistical Goodness-of-fit test
* Repetition with outcomes of 1, 2, …, 10.

**Solution:**

**Understanding Goodness-of-fit-test:**  The goodness of fit test is used to measure how effectively does an expected model fits the observed model. It’s referred to as chi-square test or ***X2 –test.*** The chi-square test is performed carrying out following operation for each of the observed and expected reading and adding them up to obtain the chi-square statistic.

Chi-Square statistic = (Observed – Expected)2 / Expected

Depending upon the degree of freedom (loosely translates to one less than the number of data point) the value thus obtained is compared with the critical chi square value obtained from the chi-square table (shown in Fig. 7.). The critical-chi square value is the X-coordinate of the chi-square distribution (shown in Fig. 8.). the Y-coordinate in the chi-square distribution indicated the probability of occurrence of a particular critical value. If the chi-square statistic is even less likely than the critical chi-square value, then its model is deemed to be a bad-fit.

  
Fig. 7. Chi-Square Table  
  
Fig. 8. Chi-Square Distribution (K is the degree of freedom)  
;

**Methodology:**  The basic working chi-square test can be done on MATLAB. The program written has comments which explains in detail the methodology that has been adopted. It is important to note that degree of freedom here is 9 although the data points are 10, this is because the last information (in the last loop) of chi\_sq\_stat can be obtained by the rest of the information.

**Program Written:**

Total\_Sample = 20000;

%random value is generated by the randi function which generates random %integer. Here, its generated in an interval.

rand\_val = randi([0 9],1,Total\_Sample);

histogram(rand\_val);

M\_data = 10;

% the unoform distribution is genrated by simply generating equal amount of

% members. The total sample is equally divided among the data points

expect\_uni\_dist = Total\_Sample/M\_data;

%the expected data set can be obtained by populating a matrix with same

%value indicating each member is sampled expect\_uni\_dist times

Expected\_data = expect\_uni\_dist \* ones(1,10);

%the observed data is captured from the histogram.

%histc captures the histogram bin counts

Observed\_data = histc(rand\_val,(0:9));

chi\_sq\_stat = 0;

for i = 1:M\_data

chi\_sq\_stat = chi\_sq\_stat + (Observed\_data(i) - expect\_uni\_dist)^2/expect\_uni\_dist;

end

%MATLAB provides an inbuilt function ‘chi2inv’ that can the provide the

% critical chi-square value for respective degree of freedom

chi\_sq\_table = chi2inv(0.95,9);

disp("Expected Data for Discrete Uniform Distributio 0,1,2,...,9");

disp(Expected\_data);

disp("Observed Data");

disp(Observed\_data);

%the final comparison is done with the chi-sqaure table entry to determine

%whther th fit is good or bad.

if(chi\_sq\_stat <= chi\_sq\_table)

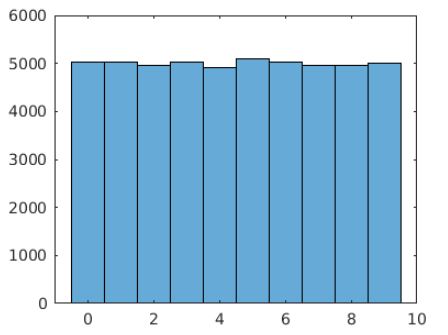
fprintf('good fit');

else

fprintf('bad fit');

end

**Results and Observations:**

****  
Fig. 9. Histogram for 50,000 Samples

Value of Member

Frequency of Sample Member

**Command Window**

Expected Data for Discrete Uniform Distribution 0,1,2,...,9  
5000 5000 5000 5000 5000 5000 5000 5000 5000 5000  
  
Observed Data  
4992 5071 4891 5067 4899 4996 5013 4990 5120 4961  
  
chi\_sq\_stat : (Chi-Sqare Statistic) <= chi\_sq\_table : (Critical Chi-Sqaure Value)   
it is a good fit

Expected Data for Discrete Uniform Distributio 1,2,...,10  
5000 5000 5000 5000 5000 5000 5000 5000 5000 5000  
  
Observed Data  
 0 5118 5073 5010 5079 4949 5027 4986 4959 4834  
  
chi\_sq\_stat : (Chi-Sqare Statistic) > chi\_sq\_table : (Critical Chi-Sqaure Value)   
it is a bad fit

Fig. 9. Shows the uniform sampled outcome of the histogram. Again, as we have large number of samples, the probability distribution obtained is uniform. In the Chi-Square test, the 3.(b) test case successfully passed the test as the Chi-Square Statistic was less than the Critical Chi-Square Value on the other hand in 3.(c) it is seen that the chi-square test failed as the Chi-Square Statistic was more than Critical Chi-Square value.