Georgia Journal of Science

Volume 62 *No. 2 Scholarly Contributions from the Membership and Others*

Article 5

2004

Demonstrating the Central Limit Theorem Using MATLAB

Kemo J. Dassau

Javier E. Hasbun
jhasbun@westga.edu

DEMONSTRATING THE CENTRAL LIMIT THEOREM USING *MATLAB*

Kemo J. Dassau
J. E. Hasbun
Department of Physics
State University of West Georgia
Carrollton GA, 30118

ABSTRACT

In this paper *MATLAB* is used in a demonstration of the Central Limit Theorem (CLT). *MATLAB* is a powerful computer program used in education and industry. *MATLAB* allows us to increase the sample size and not sacrifice speed of computation while demonstrating the basic concept of the CLT as it applies to probability and statistics. We will give its history as well as a clear understanding of its power. In addition to reproducing previous work[1], we will provide the *MATLAB* code used to perform further demonstrations. Our program will select 30 integers between one and six, as in Lazari et. al. It will then compute each individual mean (L1) and store it in a list (L5) while repeating itself n times, where n is the total number of ensembles. Upon completion, distribution plots are obtained for the n means as well as a combined histogram for each individual (L5). For a very large n, the program does indeed demonstrate that the distribution of the sample means is really normal as in Lazari et al.

Key words: Matlab, Central Limit Theorem.

INTRODUCTION

The Central Limit Theorem (CLT) states that for random samples taken from a population with a standard deviation of s (variance s^2), that is not necessarily normal (having unique values of u and s, respectively), the sampling distribution of the sample means are approximately normal when the sample size is large enough ($n \geq 35$); having a mean ($u_{\bar{x}}$) and a standard deviation s/n. In describing this relationship between sample mean and population samples, a more in depth way of stating the theorem is: The sample means of n, independent and identically distributed random variables approaches the normal distribution as n increases.

The normal distribution function originates from what is known as the probability density function (Gaussian function):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \qquad -\infty < x < \infty \tag{1}$$

This form of the density function is used as the basic format and can be transformed depending upon the actual mean (m) and standard deviation (σ) obtained from the random samples. The graph associated with the density function is known as the bell shaped curve as seen in Fig. 1.

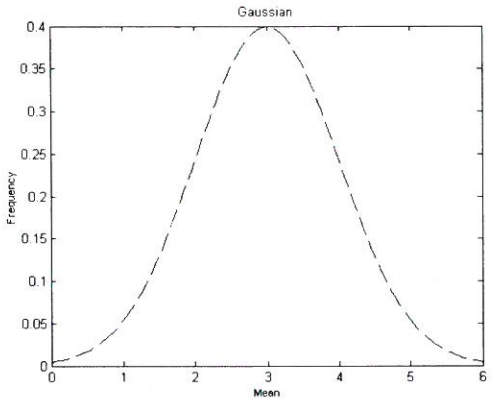

Figure 1. A typical Gaussian distribution for the case when the standard deviation equals 1.0, and the average equals 3.0.

An easier way of displaying this is by producing a frequency distribution or a histogram. The histogram's power relies on the principle that "seeing is understanding" (1). Histograms have the power of taking data and plotting frequency vs. occurrences within a range of numerical inputs. It has the capability to display not only population samples, but it can also take population sample means and graph them as well. In probability and statistics this method is used in demonstrating how as the sample size of the population means increases, the data transforms into a normal distribution.

As a result of the Central Limit Theorem, the following equations can be used to compare probabilities regarding the sample itself as well as the sample mean,

$$Z = \frac{\chi - u}{\sigma} \tag{2}$$

for the Z equation of the population, and

$$Z = \frac{\chi - u}{\sigma/n} \tag{3}$$

for the Z equation of the sample means.

These simple formulas for the Z values provide two important fundamentals of probability and statistics. First, they provide us a method of determining probabilities of occurrences given random data, using Z values and Normal Distribution Tables. Second, they also provide a more pertinent use of Z values to determine if random samples are truly normal. In order to prove the data is normal, the sample means are plotted vs. Z values. These plots are to yield a straight line as seen in Fig. 2.

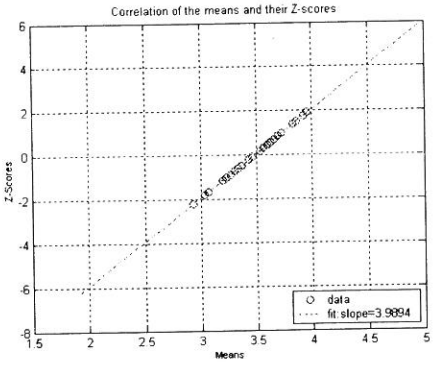

Figure 2. Plot of the sample means vs. Z scores. The straight line is indicative of the normality of data.

HISTORY

The CLT may be traced back to Pierre-Simon de Laplace and Abraham De Moivre, though their ideologies were very different.

De Moivre developed the CLT as a method for estimating discrete probabilities; in particular those involving the binomial distribution (2). De Moivre sought to determine the probability of the most frequent occurrences in a binomial distribution. He discovered that the distribution was approximately normal and could use this fact to arrive quickly at an approximation for these types of probabilities. By using this method De Moivre was then able to estimate large binomials. By using a system of factorials De Moivre was able to bypass much of the overwhelming arithmetic calculations.

On the other hand, Laplace was motivated by observational science. His ideology hinged on the understanding of determining unknown physical measurements (3). For instance, we may be interested in determining some unknown quantity, such as the maximum distance between the sun and the planets, but when the quantity is observed or measured several times, it would be highly likely that the measurements will be different, and also limited by the accuracy of the measurements (1). To determine an estimate for the unknown "constant," the natural approach is to take samples of independent measurements x_1, x_2, ...x_n and compute the mean value $X = (x_1 + x_2 + x_n)/n$. Even then, if another person repeats the process, i.e. chooses a sample, another sample of *n* measurements and calculates the means, it is highly unlikely the new measurements will produce the same means as the previous person's measurements (1). Thus, the fundamental dilemma the natural scientists were faced with was how to determine a physical quantity. As mentioned by Lazari et al., Laplace discovered that the means are distributed approximately according to the normal curve and the CLT is sometimes known as the "normal law of random errors." From these approximations Laplace was able only to state the high probability of sample means lying within a given range according to the normal distribution.

This is one of the most famous theorems of probability and statistics. Prior to the discovery of this theorem, mathematicians treated Probability and Statistics as two separate entities; however, the CLT serves as an illustration that unifies these two disciplines.

Examples

Suppose we roll a six-sided die 60 times. We should expect about an equal distribution of each of the possible outcomes. The computer program and *MATLAB* allows us to simulate this experiment easily without sacrificing time and efficiency.

Here is the equation for an output of 60 random variable between 1 and 6

EDU>> L1=round (1+5*rand (4,15)) (4)

Results of the 60 random outcomes are organized in a 4 X 15 array

L1 =

5	3	5	5	4	2	1	5	1	4	4	5	4	3	1
6	2	3	4	2	1	5	4	5	2	5	2	4	3	5
3	2	3	2	4	2	2	2	5	4	4	5	5	4	4
3	4	2	6	4	2	1	3	2	3	6	2	3	2	2

The mean is obtained by typing
EDU>> mean (L1)
Ans = 3.533

An important theory to understand is that a computer or a calculator does not produce truly random outcomes; in actuality, they use deterministic algorithms to produce strings of numbers that appear random and, in fact, may pass a variety of tests for randomness (4). It is extremely important to understand the issue of how "random" a particular random number generator is. Computers have program-based generators that provide "pseudorandom" strings of data that are used routinely for applications.

Knowing this information, let us now look at a more in depth manner of demonstrating the Central Limit Theorem. Suppose we desire to roll a 6-sixed, fair die, 30 times, while repeating the process 40 trials. Given that our successive 30 roll sequences are random, and that we can display and calculate population means, criteria for demonstrating the CLT is met, but we must ask ourselves several questions:

1. What will the population histogram resemble upon completion?
2. What will the means likely turn out to be?
3. What are the likely results for the histogram of the means?
4. Does this support real life application?

The *MATLAB* program shown in Appendix A executes a loop that rolls the die for us, N times. Each time it selects 30 random integers between 1 and 6, computes their mean and plots them on a histogram. Upon each 30 "roll" sequence an ensemble is created for students to see (Fig. 3). In each iteration, the program stores each mean in a list called, L5. It plots the distributions, L5, and as the number or ensembles increase the distribution of the means begins to shape into the familiar bell curve progressively. In L5, there will be N means. At the end, the script tests whether the distribution of the means is normal by plotting the mean's Z scores versus the means (Fig. 2). We use L5 and L1 in our program to show the ease of conversion of the TI-83 system to the *MATLAB* application.

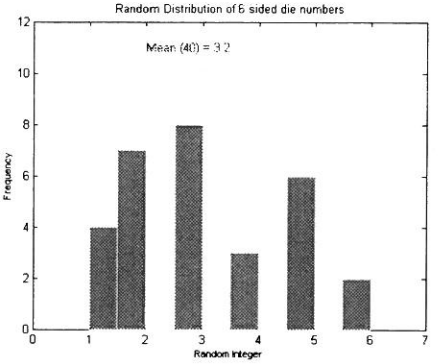

Figure 3. Shows one of many possible histograms in a set of ensembles. Each ensemble is characterized by a mean and a particular distribution.

To illustrate this powerful computer program in action we have executed it with a sample size $N = 40$. *MATLAB* will roll the die 30 times for each individual mean, and while storing the 40 means in L5, flashes the histogram for each ensemble. Each histogram will be displayed simultaneously, as each L5 is stored and displayed. When N reaches 40, the distribution or means is shown as in Fig. 4. As displayed, the resulting histogram of the L5 list represents the familiar bell shaped of the normal distribution.

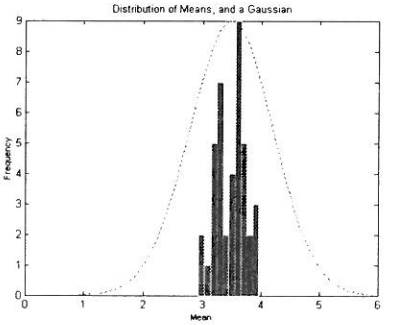

Figure 4. Shown is the distribution or means for N ensembles. The familiar bell shaped curve of the normal distribution appears as the number of ensembles increases. The number of ensembles used here is N=40.

To test whether or not the sample means are really normal we conduct the Normal Probability Plot using *MATLAB's* tools. We plot our Data List (L5) vs. the Standard Normal Variable *(Z)* as shown in Fig. 2. Since the graph is approximately a straight line, the distribution of data is approximately normal.

CONCLUSION

We have used a *MATLAB* program that allows the user to demonstrate the significance of the Central Limit Theorem. Many corporate offices, while invoking Six Sigma projects into their respective corporations, use the CLT. Bottle manufacturers gather statistical data within their manufacturing operations to ensure that they are distributing correct amounts of product to their customers (5). Health inspections in a variety of food manufacturers, such as diary products and beef manufacturers utilize information that can be deduced from the CLT to provide statistical results of bacteria and contaminations within those industries. Federal regulations state that they must meet certain requirements (within confidence intervals) in order to maintain their respective operations. This approach to the Central Limit Theorem can also be used in the classroom as an activity with discussion to help students discover and have a much better understanding of this critical statistical theorem. Because of its powerful tools, *MATLAB* is capable of performing simulations of the CLT in a very efficient way.

REFERENCES

1. Lazari A, Goel S and Kicey C: "Demonstrating the Central Limit Theorem Using a TI-83 Calculator" Georgia Journal of Science, Vol 60(3), 164-169, 2002.
2. Davis O and Goldsmith P: "Frequency Distributions." Statistical Methods in Research and Production, 9-30, 1972.
3. Myers R, Myers S, Walpole R and Keying Y: "Normal Probability Plots," Probability & Statistics for Engineers and Scientists, 143,206, 2002.
4. Chao L: "The Standard Normal Distribution" Statistics: An Intuitive Approach, 138-148, 1974.
5. Chowdbury S: "The Power of Six Sigma," 19-72, 2001 (Chicago: Dearborn Trade).

APPENDIX I

Below is the Matlab code used to perform the application of the Central Limit Theorem. The code is compatible with Matlab versions 11,12,13 and 14. Before running the code, the text needs to be saved into an "m" file. An "m" file is simply a text file, for example "clt.m". It is run under Matlab by placing the file into Matlab's working directory and by typing the name "clt" within Matlab. The program will prompt the user for input. An example of the input is N=40, so that when prompted, the number 40 is typed within Matlab.

```
% Program started by J. Hasbun and Kemo Dassau during the Fall 2003
% Program CLT converted to MATLAB from the Central limit theorem on TI 83
  calculator
% of Lazari, Goel, and Kicey
clear;
N_die=6; % 6 sided die
N_roll=30; % 30 roll ensembles
range=[1,N_die];
disp('Input N')
N = input ('N: '); % total ensembles
subplot (2,2,1)
for m=1:N;
```

```
% L1=randint (1,N_roll, range); % first way to get random numbers
 L1=round (1+(N_die-1)*rand (N_roll,1)); % second way for random numbers
% hist(y,nb) draws a histogram with nb bins.
hist(L1)
axis ([0 7 0 12]) % sets scaling for the x- and y
title ('Random Distribution of 6 sided die numbers')
xlabel('Random Integer','FontSize',8)
ylabel('Frequency','FontSize',8)
%num2str (m,p) converts the value of m to characters with p digits, cat(2,'a','b')
    concatenates a and b
L5 (m)=mean (L1);% contains the mean of the L1 distribution
str=cat(2,'Mean (',num2str(m,4),') = ',num2str(L5(m),4));
%place the the value of the time and its mean on the screen, in red
text(2,11,str,'Color','b');
%Change the color of the graph so that the bins are red and the edges of the bins are
    white.
h = findobj(gca,'Type','Patch'); % change bin colors
set(h,'FaceColor','r','EdgeColor','w')
pause(0.5);
end
%L5; % at this point L5 is a distribution of means
%figure
%pause(2)
subplot(2,2,2)
hist(L5); % histogram
title('Distribution of Means, and a Gaussian')
xlabel('Mean','FontSize',8)
ylabel('Frequency','FontSize',8)
h = findobj(gca,'Type','Patch'); % change bin colors
set(h,'FaceColor','b','EdgeColor','w')
hold on
ML=mean(L5);% this is the mean of the means
ST=std(L5); % standard deviation of the means
a=min(L5)-2.0;
b=max(L5)+2.0;
dx = (b-a)/100.0;
x = [a:dx:b];
y =max(hist(L5))*exp(-(x-ML).^2);
plot (x,y,'m:')
% We next find the z-scores of each mean and plot these versus the means
% We should get a straight line for a normal distribution => positive correlation
pause(1)
%figure
subplot(2,2,3)
z=(L5-ML)/ST; % z-scores
% Also let's superimpose a line on this graph
a=min(L5)-1;
b=max(L5)+1;
dx=(b-a)/100.0;
x = [a:dx:b];
```

```
par=polyfit(L5,z,1);% actual line fit parameters
str=cat(2,'fit:slope=',num2str(par(1))); %num2str(t) converts the value of par91) to
    characters, cat(2,'a','b') concatenates a and b
y=polyval(par,x);% actual line fit
plot (L5,z,'bo',x,y,'r:') % the z-scores plotted versus the means
legend('data',str,4);
grid on
title('Correlation of the means and their Z-scores')
xlabel('Means','FontSize',8)
ylabel('Z-Scores','FontSize',8)
h = findobj(gca,'Type','Patch'); % change bin colors
set(h,'FaceColor','b','EdgeColor','w')
```