Simulating Sampling Distribution of the Mean in R

Leslie Chandrakantha
lchandra@jjay.cuny.edu
Department of Mathematics & Computer Science
John Jay College of Criminal Justice of CUNY
USA.

ABSTRACT: *This paper presents a simulation method using a programming language called R to help students understand the concepts of sampling distributions and the central limit theorem. We simulate an approximation of a sampling distribution by taking 1000 random samples from different populations, calculating the mean of each sample, and creating histograms to display the distributions of the sample mean. Normal probability plots of the sample mean are also created as a second tool for understanding the distribution of the sample mean. The sampling distribution of the sample proportion is also studied for varying sample sizes. Students will observe the effects of sample size on the shape and spread of the approximate sampling distribution by varying the sample size*.

1. INTRODUCTION

Students in statistics classes often struggle to understand fundamental concepts such as sampling distributions, central limit theorem, confidence intervals, and hypothesis testing. Instead of understanding fundamental concepts and applying statistical procedures properly, many students focus on memorizing methods of performing calculations using calculators or software. This approach does not provide a good foundation for their future courses, conducting research, analyzing data, and making correct conclusions. The traditional way of teaching using book and lecture based instruction does not give a good understanding of the concepts to many students. Advances in technology have enabled instructors to experiment with different teaching methods. Simulations, along with the help of computers can be a very effective tool in gaining a good grasp of these concepts. One of the most challenging aspects of teaching and learning statistics is that many statistical concepts are based on the issue of what would happen if a random process such as random sampling from a population were to be repeated a large number of times. This abstract notion is very difficult for many students to grasp. Technology provides the opportunity to make this abstract idea more concrete by enabling students to repeat such random processes a very large number of times and describe their observations first hand. Furthermore, it is difficult for students in introductory level courses to understand the importance of technical conditions, such as normality assumptions and sample size requirements. Technology can be used to verify that the given data follow certain distributions. The simulation can be an effective tool in identifying the necessary sample sizes for certain situations. The *p*-value concept of hypothesis testing is also challenging for students to undersand. The simulation approach can be used to repeat the sampling process and estimate the *p*-value percentage to visualize and understand the concept. In many statistics lessons, we can use simulations to perform these types of experiments.

The American Statistical Association has published *the Guidelines for Assessment and Instruction in Statistics Education (GAISE)* [7] in order to improve learning. These guidelines recommend the active learning of concepts as an approach to teach and learn statistics. Simulations performed both manually and using computers are recommended in the GAISE report as a useful tool. In recent years, there has been much interest in the use of simulations to teach fundamental statistical concepts. Mills [10] has given a comprehensive review of the literature of computer simulation methods used in all areas of statistics to help students understand difficult concepts. Cobb [5] noted that incorporating computer simulation techniques to illustrate the key concepts and to allow students to discover important principles themselves enhances their knowledge. For more examples of the use of simulation to teach statistics see [1], [2], [3], [4], and [8].

R is now a popular tool for statistics education. Many introductory and higher level statistics instructors are now using R to teach and perform statistical calculation. Although it is a bit challenging to write statements in the command line, R can be used in conducting simulation effectively. R can easily generate random samples from a variety of probability distributions. Hallgren [9] has used R in data analysis, estimating statistical power, and constructing confidence intervals of parameters through bootstrapping. He noted that simulation methods are flexible and can be applied to a number of problems to obtain answers that may not be possible to derive through other approaches. A valuable introduction to R for introductory statistics is given in Dalgaard [6].

In this paper, we describe how to use R in an introductory statistics class to understand the concepts of sampling distribution of the mean. We generate random samples from different types of populations including more skewed distributions to study the properties of the central limit theorem. R codes are used to simulate this process for different sample sizes. Furthermore, we investigate the properties of sampling distribution of sample proportion through simulation process. Results of a survey of students' opinions on using the simulation approach and the comparison of two approaches of presenting course materials are also given.

In section 2, we give a brief overview of R and its' commands. In sections 3 and 4, we demonstrate the simulation of sampling distribution of the mean and the proportion. Section 5 gives a survey of student opinions on the simulation method and a comparison of two methods of presenting course materials. We end the paper in section 6 with some concluding remarks.

2. BRIEF OVERVIEW OF R AND SOME R COMMANDS

R is a free software environment used for working with data. R can be used to create sophisticated graphs, carry out statistical analyses, and run simulations. It is also a programming language with a set of built-in-functions. With some knowledge on coding, students can write their own codes for statistical computations. For computationally intensive tasks, one can incorporate functions written in others languages such as C, C++, and FORTRAN. R compiles and runs on Windows, MacOS, and a wide variety of UNIX platforms. The examples of R used in this paper come from the most recent version of R , R 3.4.2. R is available from http://www.r-project.org. To install R 3.4.2 on your operating system, download R from the site above using the closest mirror site to your location and choose the appropriate link for your operating system.

R is a relatively simple syntax-driven and case-sensitive language. Even though the syntax for writing instructions may be somewhat difficult initially, most students with little or no prior programming experience have become comfortable using R. R is an object-oriented program that works with data structures such as vectors (one dimensional array) and data frames (two dimensional arrays). A vector contains a list of values. When R is started, we will see a window that is called the R console. This is where we type our commands and see the text results. Graphics appear in a separate window. The > is called the prompt, where R commands are written. To quit R we type > q( ).

R can be used as a calculator. At the prompt, we enter the mathematical expression and by hitting "enter", it will calculate the result and display it. The standard arithmetic operators '+, -, *,' and '/' are used in expressions and '^' is used for exponentiation. The following example demonstrates this:

```
> 2*3 -10
[1] -4
```

The results of a calculation can be assigned to a variable (object in R) using <- or =. In this paper, we will use <-. Even though we can work with single numbers (scalars), R is primarily designed to work with vectors and functions. In R, a vector is a sequence of data values of the same type. The function, c, is used to create vectors from scalars. The following statement creates a vector:

```
> x <- c(2,4,6,8,0)
> x
[1] 2 4 6 8 10
```

Once we have a vector of numbers, we can apply built-in functions to get useful statistical summaries and visual displays. R also provides functions for generating random samples from various probability distributions.

Control Structures

R has the standard control structures such as *if, while,* and *for,* which can be used to control the flow of an R code. We will demonstrate the use of control structures in R using the following code segment. Let's assume that we have stored 1000 numbers in the vector named x. The following code will compute the average of the nonnegative numbers in vector x. The symbol # is used to write comments.

```
> total <- 0             # variable total initialized to 0
> count <- 0             # variable count initialized to 0
> for(i in 1:1000)
+  {
+    if (x[i] >= 0)
+     {
+      count <- count +1    # count the positive values
+      total <- total + x[i]    # add the positive values
+     }
+  }
> average <- total/count  # compute the average
```

In the above code segment, the *for* loop iterates 1000 times, selecting only nonnegative numbers using the if statement. It computes the average as well. The variable named *count* counts the number of nonnegative numbers stored in x. We will use the control structures when we discuss simulations in the following sections.

3. SIMULATION OF SAMPLING DISTRIBUTION OF MEAN

It is important that students understand the concepts behind sampling distributions and how they apply towards making statistical inferences. Sampling distributions are important because inferential statistics are based on them. Inferential statistics is about drawing conclusions about the population based on sample data. The sampling distribution of the mean is the probability distribution of the sample mean based on all possible simple random samples of the same size from the same population. The sampling distribution of the mean has the following properties:

- The mean of all sample means, $\mu_{\bar{x}}$, is equal to the population mean μ.
- The standard deviation of the sample means, $\sigma_{\bar{x}}$, (known as the standard error) is equal to the population standard deviation σ divided by square root of the sample size *n*.

- The distribution of the sample mean $\overline{x}$ is more normal than the distribution of individual observations *x*.

The central limit theorem explains the shape of the sampling distribution. This theorem tells that for a population of any distribution, the distribution of the sample mean approaches a normal distribution as the sample size increases. The larger the sample size, the better the approximation. Based on this theorem, we can use the normal distribution for inferences about the mean for larger sample sizes, even if the original population is not normally distributed. Many students use this fact without understanding the underlying concept but by having the simulation students can digest the material and concepts to a better extent.

Now we demonstrate the simulation of the sampling distribution using R. R is installed on the school computers which the students use to follow along during the class time. We begin by considering different population distributions and different sample sizes to observe the effects of the sample size and the shape of the original distribution on the sampling distribution of the mean. The uniform, chi-square, and normal populations and sample sizes of 10, 25, and 50 values are considered. These three populations have uniform, skewed, and bell shapes so each student can visualize how as the sample size increases as well and the sampling distribution approximates a normal distribution for different original shapes. This is one of the main points that we want to teach. These three populations, their shapes, and parameters were explained in previous lessons. The R functions are then introduced to generate random variants from these three populations. *Table 1* gives the characteristics of the samples and the R functions. The value of n in R functions is the sample size.

Table 1: Sample Characteristics

Distribution	**Sample Sizes**	**Mean**	**Std Deviation**	**R Function**
Uniform (0, 100)	10, 25, 50	50	28.87	= runif(n, 0, 100)
Normal (100, 10)	10, 25, 50	100	10	= rnorm(n, 100, 10)
Chi-square (2)	10, 25, 50	2	2	= rchisq(n, 2)

Now we show the R codes for generating random samples of 10 uniform random numbers and computing the sampling distribution of the mean. The following code segment generates 1000 random samples sizes of 10 values from the above uniform distribution and proceeds to compute the sample means. The R function *runif* is used to generate a random sample from the uniform distribution. The syntax of the function is *runif(n, a, b).* This function returns a random sample size of n values from the uniform distribution from *a* to *b*.

```
> means <- c()
> for(i in 1:1000)
+  {
+    y <- runif(10,0,100)
+    means[i] <- mean(y)
+  }
> mean(means)
[1] 49.75268
> sd(means)
[1] 9.164981
> hist(means, main = "U(0,100), n=10")
> qqnorm(means, main = “U(0,100), n=10”)
```

The *for* loop calculates means and also iterates 1000 times to generate 1000 random samples from the uniform distribution. The *means* vector holds these sample means. The *runif(10,0,100)* function generates a random sample size of 10 values from the uniform distribution from 0 to 100. This random sample is stored in vector y for the *i*th iteration of the *for* loop. The *means[i]* variable stores the corresponding sample mean for each sample for the *i*th iteration. The *mean* and *sd* commands compute the mean and standard deviation of 1000 sample means. Those 1000 sample means are then considered as the approximate sampling distribution of the mean to verify the validity of the properties the mean and standard deviation (standard error) of the sampling distribution. To study the shape of the sampling distribution, we create a histogram of the 1000 sample means using the *hist* command. A better way of deciding if data is normally distributed is by creating a normal probability plot. In R, we can create a normal probability plot using *qqnorm* command. In normal probability plot, data are plotted against a theoretical normal distribution in such a way that the points should ideally form a straight line. Departures from a straight line indicate departures from normality.

Similarly, we generate the approximate sampling distributions, histograms, and normal probability plots for all the cases we have considered in this lesson. *Figure 1* and *Figure 2* show histograms and normal probability plots for all the cases. In both figures, the first, second, and third column histograms and normal probability plots are created from sample means from uniform, normal, and chi-square distributions respectively. The first, second, and third rows represent samples sizes 10, 25, and 50 values respectively. These histograms allow students to understand the meaning of the central limit theorem. Students will be able to visualize that as the sample size increases, the shape of the distribution is becoming more normal and therefore the sample means are less variable. For skewed data such as chi-square data, the sampling distribution does not approach a normal distribution for sample sizes as large as 50 values.

To investigate the effect of the sample size on the shape of more skewed distributions, we consider the lognormal distribution. The lognormal distribution is one of the heavy tail distributions. The probability density function (pdf) of the lognormal distribution with $\mu = 0$ and $\sigma = 1$ is created using R and shown in *Figure 3*. *Figure 4* shows the histograms of sample means each from 1000 random samples with sample sizes 50, 100, and 500 respectively. From *Figure 4*, students understand that for more skewed distributions, sampling distribution does not approach a normal distribution in the case of sample sizes as large as 500.

In the case of a symmetrical uniform distribution, the sampling distribution can be approximated by a normal distribution for a smaller sample size such as 10 values. These facts can also be seen through normal probability plots. The effect of the sample size (n) on the standard error ($\sigma/\sqrt{n}$) of the sampling distribution is also evident from the histograms. Students will notice that as sample size increases, the spread of the sampling distribution decreases. At the end of the lesson, students are able to comprehend the central limit theorem and understand the properties of the sampling distribution discussed in the class. The R codes used in this paper are given in the Appendix.

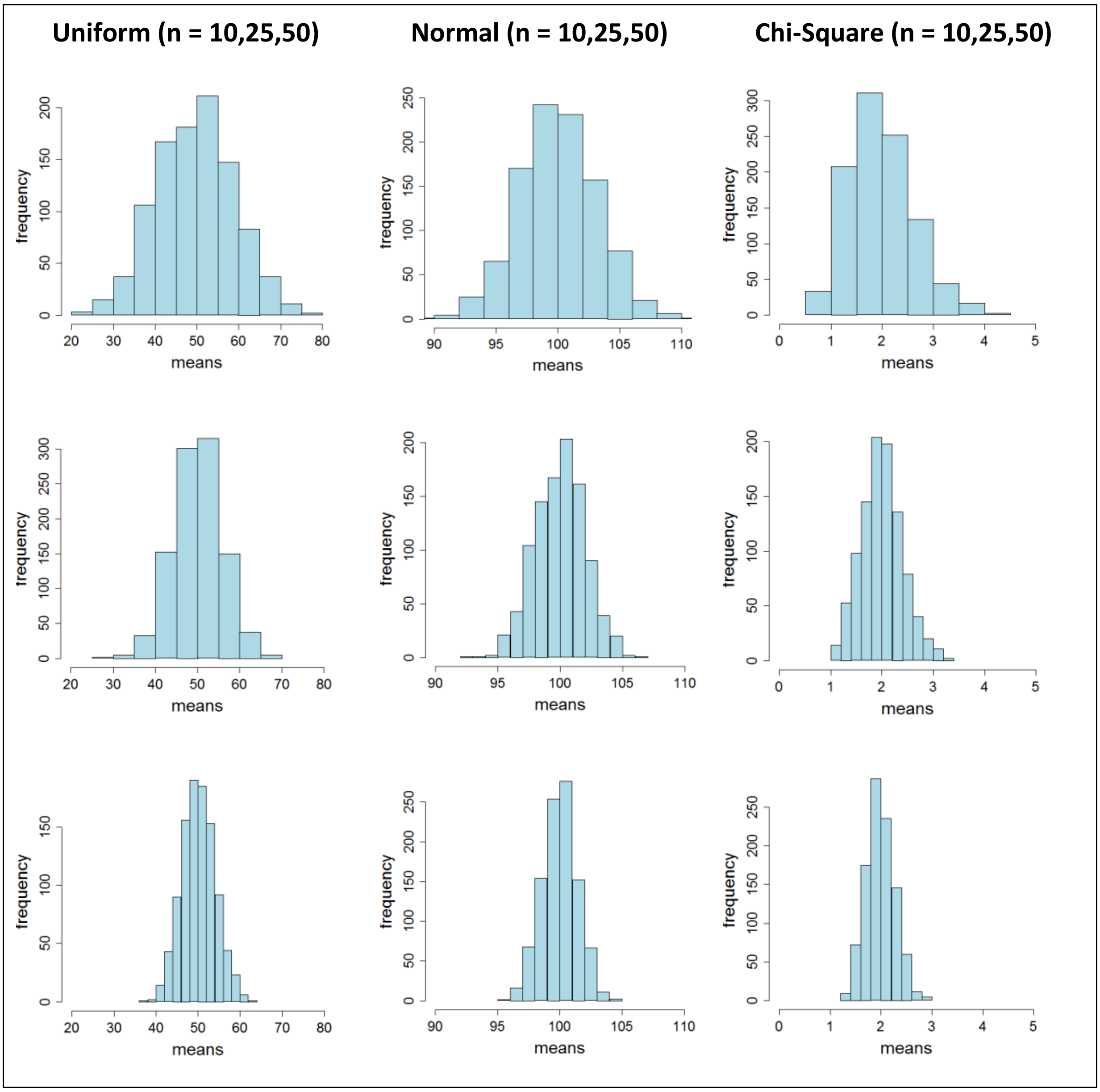

Figure 1: Histograms for Sample Means

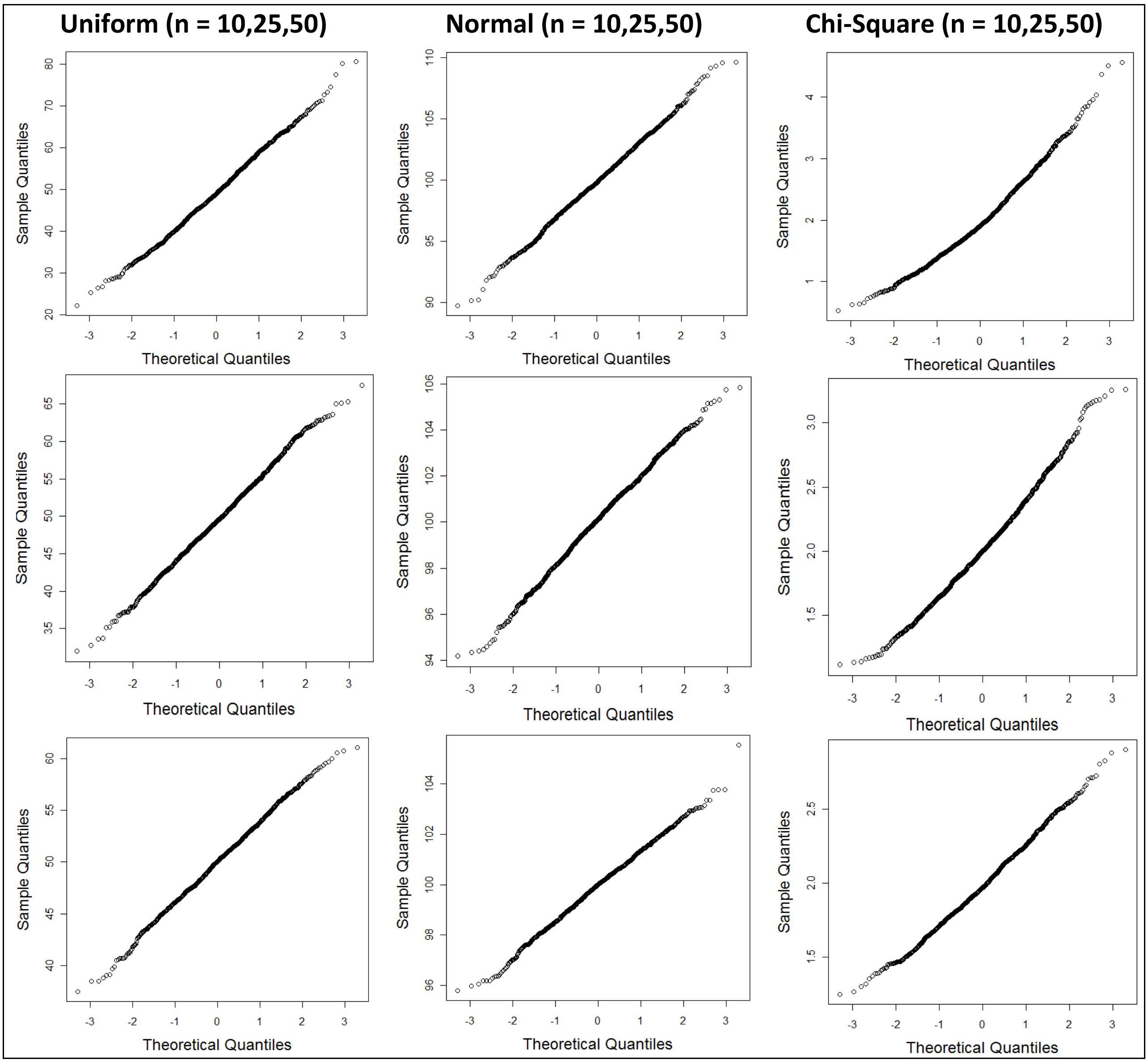

Figure 2: Normal Probability Plots for Sample Means

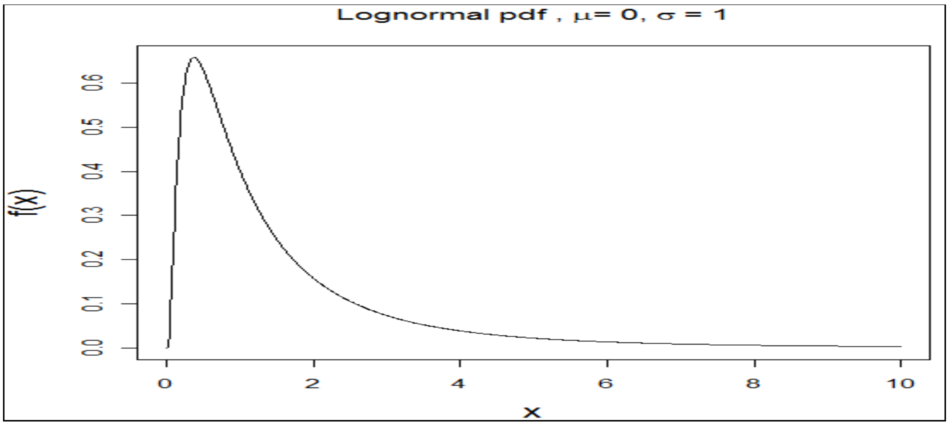

Figure 3: Lognormal Probability Density Function

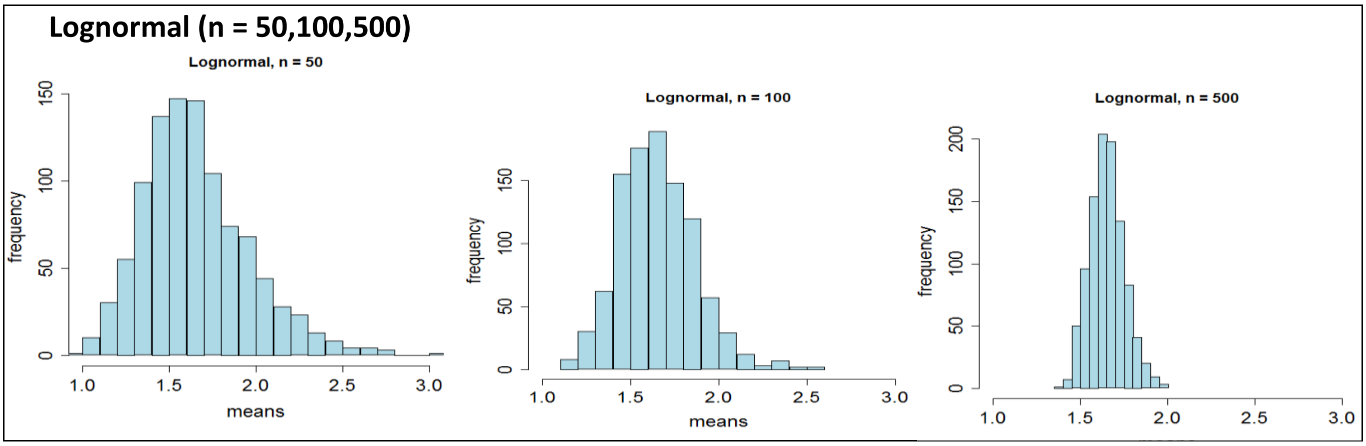

Figure 4: Histograms for Sample Means from Lognormal Distribution

4. SAMPLING DISTRIBUTION OF SAMPLE PROPORTION

Sample proportion is commonly used in introductory statistics classes. In this section, we examine the sampling distribution of the sample proportion using the simulation. The population proportion, denoted by *p*, is obtained by taking the ratio of the number of elements in a population with a specific characteristic to the total number of elements in the population. The sample proportion, denoted by $\hat{p}$, gives a similar ratio for a sample. Just like the sample mean, the sample proportion is a random variable. The sample proportion can assume one of a large number of possible values depending on the sample. The probability distribution of the sample proportion is called the sampling distribution of the sample proportion. The mean of the sampling distribution is equal to the population proportion, *p*, and the standard deviation is given by $\sqrt{\frac{p(1-p)}{n}}$. The central limit theorem for sample proportion approximates the sampling distribution by a normal distribution for a sufficiently large sample size. The sample size is considered to be sufficiently large if both *np* and *n(1-p)* are at least 15. Students sometimes use these limits without understanding the meaning behind it. The following simulation process will give a better understanding of the meaning.

We generate approximate sampling distributions and create histograms of the sample proportion $\hat{p}$ for *p* = 0.1, 0.25, and 0.5, and *n* = 25, 50, 100, and 200. *Figure 5* shows the histograms for these values of *p* and *n*. In the figure, the first, second and third column histograms represent *p* values 0.1, 0.25, and 0.5 respectively. The first, second, third and fourth rows represent sample sizes 25, 50, 100, and 200 respectively. These histograms show that as sample size increases, the sampling distribution becomes more normal. For smaller values of *p*, larger sample is needed to approximate the sampling distribution by a normal distribution. It is easy to notice that both *np* and *n(1-p)* are larger for *p* values closer to 0.5. For *p* = 0.5, a sample size of *n* = 50 is enough to approximate the sampling distribution by a normal distribution. When *p* gets closer to 0.1 (or to 0.9), sample sizes as large as 100 may not be enough for the normal approximation. This simulation lesson allows student to understand the meaning of this concept.

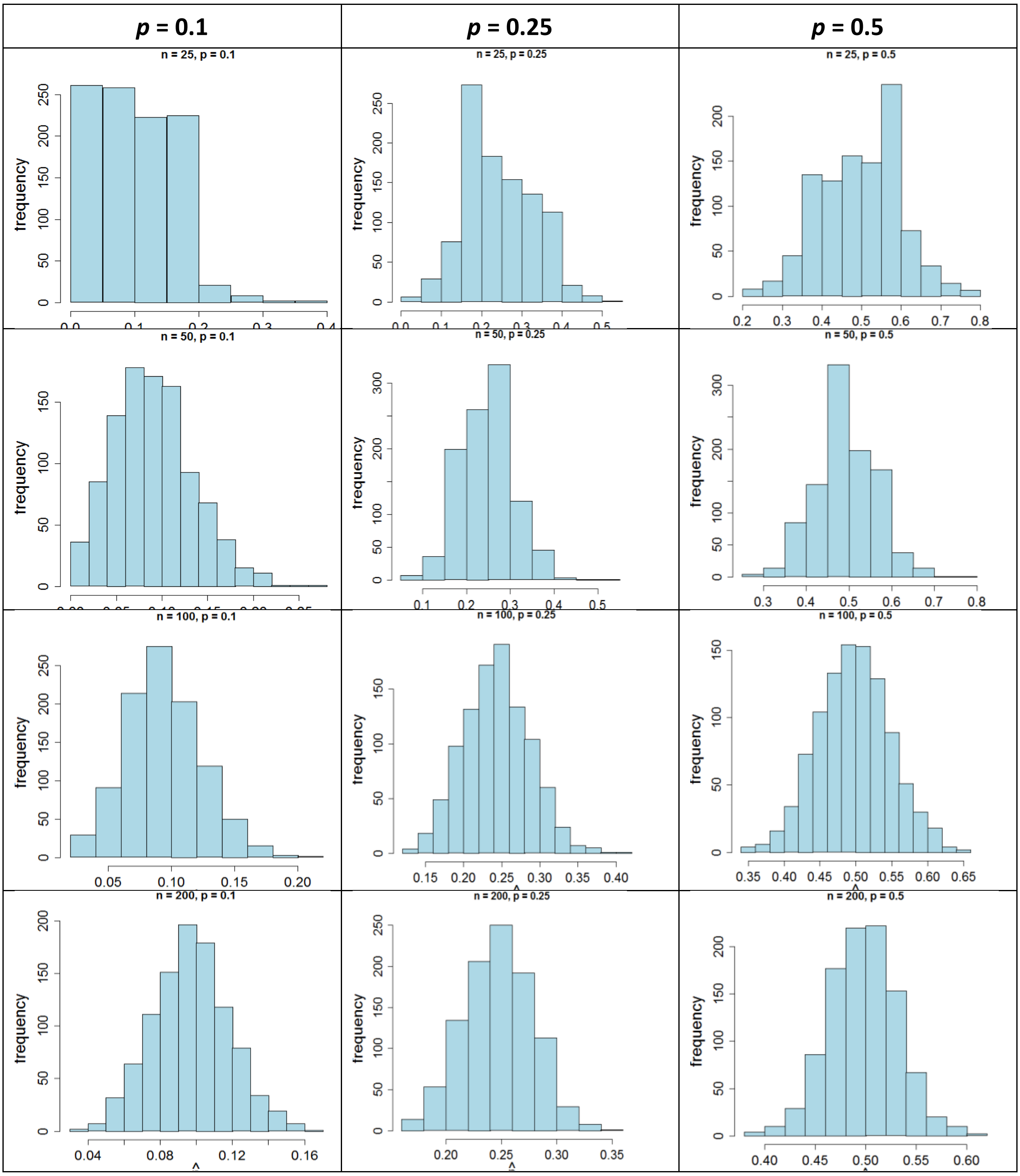

Figure 5: Histograms for Sample Proportions

5. STUDENT OPINION AND COMPARISON OF TWO METHODS

This simulation approach was introduced in the probability and statistics course offered in the Mathematics Department. Most students taking this course are either sophomores or juniors majoring in Computer Information Systems, General Sciences, and other quantitative disciplines. These students have taken at least one computer programming course previously or have been exposed to some sort of computer logic, depending on their discipline. In Fall 2016, we introduced simulation methods for the first time. In previous semesters, we taught the same material without using simulation methods. To evaluate the effectiveness of simulation methods on understanding the concepts, we compare Fall 2016 grades with the Spring 2016 grades of the test related to sampling distributions and central limit theorem. The test questions are not the same within the two semesters, but they are very similar. *Table 2* gives the test scores statistics below:

Table 2: Test Scores Statistics

Class	*n*	Mean	Median	Std. Dev.
Fall 2016	26	77.34	73	14.25
Spring 2016	36	70.45	68	16.41

Using R, the two sample *t*-test was performed on test scores to test the hypothesis that the Fall 2016 class (using simulation methods) performs better on average than the Spring 2016 class (not using simulation methods). The *p*-value produced by R was 0.0451. This *p*-value indicates that the Fall 2016 class performed better at a 0.05 significance level.

The simulation approach was used by the Probability and Statistics class (Fall 2016) to teach the sampling distributions lessons. Afterwards, the following survey was conducted in class to measure students' opinion on simulation methods. All 26 students answered the survey. We have not conducted the survey in previous semesters. Each question has three answering options, namely yes, no, and no opinion. As we mentioned earlier, these students were majoring in Computer Information Systems, sciences, or other quantitative disciplines. Most of them are either sophomores or juniors. *Table 3* gives the summary of their responses.

1. Method helps me to understand the concepts.
2. Feeling like I am part of the discussion.
3. Feeling comfortable taking part in the lesson.
4. Visual representation of outcomes is useful in understanding.
5. Recommend this approach to other students.

We selected these questions to learn the students' reflection on their own learning of the concepts and the opinion on simulation approach in presenting course material. As we mentioned earlier in this paper, the American Statistical Association's *GAISE* report [7] has recommended several key points to improve student learning in statistics, including fostering active learning and the use of technology to explore concepts and analyze data. Answers to these survey questions will reveal students' perception of the methods that help to achieve those goals. At the same time, we can identify the effectiveness of this approach in terms of students' opinions.

Table 3: Summary of Responses

Question	Yes	No	No Opinion
1	94%	0%	6%
2	52%	12%	36%
3	58%	18%	24%
4	85%	3%	12%
5	76%	9%	15%

The majority of the students answered yes to all five questions. Looking at the percentages, we observe that students believe they understand the concepts better with this approach. A higher majority of students answered yes to questions 1, 4, and 5, which indicates that visual explanations of concepts enhance learning. To get a clear understanding about the percentages, we performed ztests for each question. At a 5% significance level, we tested the claim that the proportion of students who answered yes to each question was significantly greater than 50%. *Table 4* gives the *p*-values for each question. Notice from the *p*-values for questions 1, 4, and 5, the proportions of students answered yes are significantly greater than 50%. Even if we test against 60%, the proportions for these three questions are significantly greater. This is the same conclusion we made by looking at the row percentages. The proportions for questions 2 and 3 are not significantly greater than 50%. The instructor should motivate students to join the lesson enthusiatcally and make them more comfortable during the lesson.

Table 4: *p*-values

Question	*p*-value
Q1	0.000
Q2	0.419
Q3	0.204
Q4	0.000
Q5	0.001

We have to take caution in that the sample sizes for the survey and assessment statistics provided are not large enough to make a firm judgment on the conclusion. We plan to use larger sample sizes in the future in order to give a comprehensive survey and to make a formal assessment.

6. CONCLUSION

Many students struggle with understanding statistics certain concepts such as sampling distributions. Simulations can be effective learning tools in helping students understand abstract concepts associated with repeated random processes. We have demonstrated the use of simulations by using R to teach these topics. This is a very useful way to visualize and understand the sampling distribution and the central limit theorem. These simulation methods accommodate students who have a various background in mathematics. More empirical studies need to be conducted to measure the effectiveness of using simulations as a pedagogical tool.

7. REFERENCES

[1] Barr, Graham D. and Scott, Leanne. (2011). Teaching Statistics in a Spreadsheet Environment using Simulation. *Spreadsheets in Education* (eJSiE), 4(3). http://epublications.bond.edu.edu.au/ejsie/vol4/iss3/2.

[2] Butler, A., Rothery, P., & Roy, D. (2003). Minitab Macros for Resampling Methods. *Teaching Statistics*, 25 (1), 22-25.

[3] Chandrakantha, Leslie. (2014), Visualizing and Understanding Confidence Intervals and Hypothesis Testing Using Excel Simulation. *The Electronic Journal of Mathematics and Technology* (EJMT), 8(3): p 212-221

[4] Christie, D. (2004). Resampling with Excel. *Teaching Statistics*, 26 (1), 9-14.

[5] Cobb, P. (1994). Where is the Mind? Constructivist and Sociocultural Perspectives on Mathematical Development. *Educational Researcher*, 23, 13-20.

[6] Dalgaard, P., *Introductory Statistics with R*, 2nd Edition, New York, NY: Springer, 2008.

[7] GAISE (2005). Guidelines for Assessment and Instruction in Statistics Education Report. American Statistical Association, Alexandria, VA. http://www.amstat.org/education/gaise/

[8] Hagtvedt, R., Jones, G. T., & Jones, K. (2008). Teaching Confidence Intervals Using Simulation. *Teaching Statistics*, 30 (2), 53-56.

[9] Hallgren, K. A., (2103). Conducting Simulation Studies in the R Programming Environment. *Tutorial in Quantitative Methods for Psychology*, 9(2), 43-60.

[10] Mills, J. D. (2002). Using Computer Simulation Methods to Teach Statistics: A Review of the Literature. *Journal of Statistics Education (Online)*, 10 (1). http://www.amstat.org/publications/jse/v10n1/mills.html

Appendix

The following R codes were used in the paper:

Computing sampling distribution from normal distribution

```
> means  <-  c()
> for (i in 1:1000)
+ {
+   y <- rnorm(10,100,10)
+   means[i] <- mean(y)
+ }
> mean(means)
[1] 99.91111
> sd(means)
[1] 3.086187
> hist(means, main = "N(100,10), n = 10")
> qqnorm(means,main = "N(100,10), n = 10")
```

Computing sampling distribution from Chi-square distribution

```
> means <- c()
> for(i in 1:1000)
+  {
+   y <- rchisq(10,2)
+   means[i] <- mean(y)
+  }
> mean(means)
[1] 2.002608
> sd(means)
[1] 0.6131161
> hist(means, main = "Chi-square(2), n = 10")
> qqnorm(means, main = "Chi-square(2), n = 10")
```

Computing sampling distribution from lognormal distribution

```
> means <- c()
> for(i in 1:1000)
+  {
+   y <- rlnorm(50,0,1)
+   means[i] <- mean(y)
+  }
> hist(means, main = "Lognormal(0,1), n = 50")
```

Creating lognormal pdf

```
> x <- seq(from = 0, to = 10, by = 0.01)
> y <- dlnorm(x,0,1)
> plot(x,y, type = "l", xlab = "x", ylab = "f(x)",main =
+ expression(paste("Lognormal pdf, ",mu, " = 0, ",sigma, " =    +
1")))
```

Computing sampling distribution for proportion

```
> phat <- c()
> for(i in 1:1000)
+  {
+    phat[i] <- rbinom(1,25,0.1)/25
+  }
> hist(phat, xlab = expression(hat(p)), ylab = "frequency",
+ main = "n = 25, p = 0.1")
```