AC 2010-367: USING SIMULATION AS AN AID IN UNDERSTANDING ENGINEERING STATISTICS

Anthony Richardson, University of Evansville
Using Numerical Computation Software as an Aid to Understanding Engineering Statistics

Abstract

A course in Probability and Statistics is required in most engineering programs. Many students struggle with some of the fundamental concepts in this course. Some key concepts that are difficult for students to grasp include understanding (1) what it means for random variables to have different distributions, (2) that a sample statistic is a random variable and not a number, (3) that sample statistics will typically have different distributions than observations from the population, and (4) how sample size can affect the quality of an estimate.

The use of numerical computation software to simulate statistical experiments is described. Although numerical computation software has been used in this course in the past, its use has been restricted to computing sample statistics. By requiring the students to use numerical software in order to do simulations, it was hoped that students would develop a much deeper understanding of theoretical concepts. This paper describes the many ways in which numerical analysis software was used in the course (homework and programming projects). A survey was used to assess the degree to which the software aided in understanding course concepts. The results of the survey are presented and discussed.

Introduction

Undergraduate engineering and computer science students at the University of Evansville are required to take a ½ semester course in Probability and Statistics. This course is typically taken during the Junior year. Calculus II is the only prerequisite for the course, although most students have also taken Calculus III and a Differential Equations course.

There are several computer application programs that can be used for experimental simulation. MATLAB and/or Octave was chosen primarily because most of the students taking this course already had previous experience with MATLAB. Because the course is only ½ semester in duration there is not time to teach the students how to use a new software package. Octave is free software that is highly compatible with MATLAB. Both MATLAB and Octave are installed on computers in the College of Engineering. The simulation experiments require that the Statistics Toolbox (for MATLAB) or the Statistics Package (for Octave from the Octave Forge) be installed. Most students chose to do the experiments using Octave (primarily because they could use it on their personal computers for free).

All commands and program code in this paper have been tested under Octave. They should run with few (or, in most cases, no) changes under MATLAB.

Descriptive Statistics

Octave contains many functions that are useful for the computation of standard descriptive statistics. A few of the more common functions are listed in Table 1.
Most of the functions listed take additional optional arguments that can be used to further define the behavior of the function. For example, \texttt{var(x, 0)} computes the variance of a sample using

\[ \text{var}(x, 0) = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{X})^2 \]  

(1)

where \( x \) is an array containing all observations, \( x_i \) is an observation in the sample, \( \bar{X} \) is the sample mean and \( N \) is the number of observations in the sample. (This formula gives the best unbiased estimator of the variance.) Alternatively, \texttt{var(x, 1)} computes the variance using the following formula (this is the population variance, \( \sigma \), or the second sample moment around the mean)

\[ \text{var}(x, 1) = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{X})^2. \]  

(2)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>length(x)</td>
<td>Number of observations in sample</td>
</tr>
<tr>
<td>sort(x)</td>
<td>Sort observations in increasing order</td>
</tr>
<tr>
<td>mean(x)</td>
<td>Compute the average value</td>
</tr>
<tr>
<td>median(x)</td>
<td>Compute the median value</td>
</tr>
<tr>
<td>min(x)</td>
<td>Finds the minimum value in the sample</td>
</tr>
<tr>
<td>max(x)</td>
<td>Finds the maximum value in the sample</td>
</tr>
<tr>
<td>std(x)</td>
<td>Compute the standard deviation</td>
</tr>
<tr>
<td>var(x)</td>
<td>Compute the variance</td>
</tr>
<tr>
<td>quantile(x, q)</td>
<td>Compute the q\textsuperscript{th} sample quantile</td>
</tr>
<tr>
<td>trimmean(x, p)</td>
<td>Compute mean excluding high and low p percent of data</td>
</tr>
<tr>
<td>boxplot(x)</td>
<td>Produce a box plot</td>
</tr>
<tr>
<td>hist(x)</td>
<td>Produce a histogram</td>
</tr>
</tbody>
</table>

Table 1: Octave Descriptive Statistical Functions

Figure 1 shows a boxplot\textsuperscript{1} produced using the Octave \texttt{boxplot} function. The underlying sample consists of 100 observations and are believed to have a lognormal distribution. The vertical left and right edges of the box are at the lower and upper quartile values. The vertical line inside the box is located at the sample median value. Denoting the interquartile range (the difference between the upper and lower quartiles) by \( f_s \), then the box \textit{whiskers} include all data that lie within 1.5\( f_s \) of the lower or upper quartile. Outliers (data points that are farther than 1.5\( f_s \) from the closest quartile) are marked with a '+'\textsuperscript{1}. Extreme outliers (data points that are farther than 3\( f_s \) from the closest quartile) are marked with an 'o'. The command \texttt{boxplot(x, 0, '+o', 0)} was used to create the boxplot in this figure.

Figure 2 shows a histogram for the same set of observations. The figure was produced using just
a simple `hist(x)` command. Octave automatically sets the bin widths and positions so that there are 10 histogram bins between the minimum and maximum values in the sample. Optional arguments allow the number of bins, the bin widths, and the bin positions to be specified. The histogram can also be normalized. Figure 3 shows the result of the command 
`hist(x, [1.25:2.5:30], 1/2.5)` for the same sample used in Figure 2. In this command, the second (vector) argument specifies the positions of the histogram bins. The third argument is equal to the reciprocal of the bin width and causes the resulting histogram to be normalized so that the total histogram area is equal to one.

![Figure 1: Example Boxplot](image)

**Distribution Functions**

Octave contains probability density functions (PDFs) and cumulative distribution functions (CDFs) for most of the classically defined probability distributions. A few of the available functions are listed in Table 2. (These functions are also included in the MATLAB Statistical Toolbox.)

The density functions can be used to verify whether observations have a particular distribution. The observations in the sample used to produce the histograms in Figures 2 and 3 are believed to be from a lognormal distribution with parameters \( \mu = 1 \) and \( \sigma = 1 \). Figure 4 shows a plot of the theoretical lognormal pdf overlaying the normalized histogram of the observations. The relatively good agreement between the pdf and the histogram helps to confirm the assumption that the data are lognormally distributed.

The distribution functions have much greater utility. They are most often used to determine probabilities. This will be demonstrated in some of the examples later in this paper. They can also replace the probability tables found in most statistics texts. Most tables only contain probabilities for certain parameter values and interpolation must be used to determine probabilities for non-listed values. The Octave distribution functions can be used to find probabilities for arbitrary parameter values. The distribution functions are also useful for generating probability tables for inclusion in closed-book examinations.
### Octave Function

<table>
<thead>
<tr>
<th>Distribution</th>
<th>PDF</th>
<th>CDF</th>
<th>Inverse CDF</th>
<th>RV Generation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binomial</td>
<td>binopdf(x, n, p)</td>
<td>binocdf(x, n, p)</td>
<td>binoinv(η, n, p)</td>
<td>binornd(n, p, ro, co)</td>
</tr>
<tr>
<td>Hypergeom.</td>
<td>hygepdf(x,N,M,n)</td>
<td>hygecdf(x,N,M,n)</td>
<td>hygeinv(η,N,M,n)</td>
<td>hygernd(N,M,n,ro,co)</td>
</tr>
<tr>
<td>Neg. Binom.</td>
<td>nbinpdf(x, r, p)</td>
<td>nbincdf(x, r, p)</td>
<td>nbininv(η, r, p)</td>
<td>nbinrnd(r, p, ro, co)</td>
</tr>
<tr>
<td>Poisson</td>
<td>poisspdf(x, λ)</td>
<td>poisscdf(x, λ)</td>
<td>poissinv(η, λ)</td>
<td>poissrnd(λ, ro, co)</td>
</tr>
<tr>
<td>Uniform</td>
<td>unifpdf(x, A, B)</td>
<td>unifcdf(x, A, B)</td>
<td>unifinv(η, A, B)</td>
<td>unifrnd(A, B, ro, co)</td>
</tr>
<tr>
<td>Normal</td>
<td>normpdf(x, μ, σ)</td>
<td>normcdf(x, μ, σ)</td>
<td>norminv(η, μ, σ)</td>
<td>normrnd(μ, σ, ro, co)</td>
</tr>
<tr>
<td>Exponential</td>
<td>exppdf(x, λ)</td>
<td>expcdf(x, λ)</td>
<td>expinv(η, λ)</td>
<td>exprnd(λ, ro, co)</td>
</tr>
<tr>
<td>Gamma</td>
<td>gampdf(x, α, β)</td>
<td>gamcdf(x, α, β)</td>
<td>gaminv(η, α, β)</td>
<td>gamrnd(α, β, ro, co)</td>
</tr>
<tr>
<td>Chi Squared</td>
<td>chi2pdf(x, ν)</td>
<td>chi2cdf(x, ν)</td>
<td>chi2inv(η, ν)</td>
<td>chi2rnd(v, ro, co)</td>
</tr>
<tr>
<td>Weibull</td>
<td>wblpdf(x, β, α)</td>
<td>wblcdf(x, β, α)</td>
<td>wblinv(η, β, α)</td>
<td>wblrnd(β, α, ro, co)</td>
</tr>
<tr>
<td>Lognormal</td>
<td>lognpdf(x, μ, σ)</td>
<td>logncdf(x, μ, σ)</td>
<td>logninv(η, μ, σ)</td>
<td>lognrnd(μ, σ, ro, co)</td>
</tr>
<tr>
<td>Std Beta</td>
<td>betapdf(x, α, β)</td>
<td>betacdf(x, α, β)</td>
<td>betainv(η, α, β)</td>
<td>betarnd(α, β, ro, co)</td>
</tr>
<tr>
<td>Student t</td>
<td>tpdf(t, ν)</td>
<td>tcdf(t, ν)</td>
<td>tinv(η, ν)</td>
<td>trnd(v, ro, co)</td>
</tr>
<tr>
<td>F</td>
<td>fpdf(f, v₁, v₂)</td>
<td>fcdff(f, v₁, v₂)</td>
<td>finv(η, v₁, v₂)</td>
<td>frnd(v₁, v₂, ro, co)</td>
</tr>
</tbody>
</table>

Table 2: Octave Probability Functions

The inverse cumulative distribution functions are useful for determining quantiles. For example, the median of the lognormal distribution shown in Figure 4 (with parameters \( μ = 1 \) and \( σ = 1 \)) can be computed using \texttt{logninv(0.5, 1, 1)} (with result 2.7183). The inverse functions could be used to generate random variables with the given distribution via the inversion method\(^2\). This is unnecessary in Octave, because random number generator functions exist for each distribution that has an inverse distribution function.

### Using Simulation to Estimate Probabilities

Some of the random variable generation functions that are available in Octave are listed in the last column of Table 2. These functions make it easy to simulate experiments in probability. Problem 1 demonstrates the utility of this approach.

Assume that random variable \( X \) has a Poisson distribution with parameter \( λ = 5 \). Find the probability that \( X \leq 8 \), that is, find \( P(X \leq 8) \).

**Problem 1: A Simple Problem in Probability**

The \texttt{poisscdf} cumulative distribution function can be used to find the desired theoretical result:

\[
P(X \leq 8) = \text{poisscdf}(8, 5) = 0.93191
\]

(3)
(Since \(X\) is a discrete random variable this probability could also be computed by summing values from `poisspdf`. In this case, \(P(X \leq 8) = \text{sum}(\text{poisspdf}(0:8, 5))\).)

However, we can also use simulation to approximately determine this probability. The following code uses the Poisson random number generator to create a sample of \(N = 100\) observations that are Poisson distributed with parameter \(\lambda = 5\). It then finds the relative frequency of \(X \leq 8\) as an estimate of \(P(X \leq 8)\).

```
% Estimate P(X <= 8)
N = 100; lambda = 5;
x = poissrnd(lambda, 1, N);
% sum(x <= 8) is equal to the number of values less than or equal to 8.
P = sum(x <= 8)/N
```

Program Listing 1: Estimating Probabilities for Problem 1

Simulation can help to confirm that the theoretical value obtained from the CDF is correct. It should be emphasized, though, that simulation provides only an estimate of the true value. It is easy to demonstrate that this estimate is actually a random variable simple by running the program several times. (10 example runs of the program yield values for \(P\) between 0.91 and 0.97 with mean 0.942 and standard deviation 0.0204.) The values of \(P\) are close to the theoretical value, but (typically) not equal to it. Students tend to expect the relative frequency to equal the theoretical probability. This simulation helps ease the transition from descriptive statistics to inferential statistics. It is also easy to demonstrate that increasing the number of observations yields better estimates. After changing the value of \(N\) to 1000, 10 successive runs of the program yield values of \(P\) between 0.917 and 0.942 with mean 0.928 and standard deviation 0.0079. For the larger value of \(N\), the \(P\) values are much closer to the theoretical value.

Simulation examples like this can be performed quickly in the classroom and can help to increase the student's understanding of basic probability concepts, because they appeal to the student's intuitive understanding of probability. Students should be encouraged to simulate homework problems in probability to verify their theoretical results.

Problem 2 is a more complex problem that was given as a project assignment in the course. The original problem was taken from the course text and is shown in italics. The theoretical mean and standard deviation of drill lifetime are 124 hours and 117 hours respectively. Letting \(X\) denote drill lifetime, then the theoretical values for \(P(X \leq 100)\) and \(P(X \geq 200)\) are 0.552 and 0.159 respectively. The problem can be simulated using fewer than 10 lines of Octave code (see Program Listing 2). An example run of the code gives estimates of the mean and standard deviation of 122 hours and 101 hours. Estimates of the two probabilities are 0.540 and 0.160 respectively. Again, a key point of emphasis is that these estimates are also random variables. Subsequent runs will yield slightly different estimates. The estimates can be expected to be better (smaller variance) for larger sample sizes. The project requires that the students verify this by creating histograms of the sample means for sample sizes of \(N = 100\) and \(N = 400\). Typical results are shown in Figures 5 and 6. Both sample means are unbiased estimates of the true mean and it is apparent that the sample mean with \(N = 400\) has much lower variance and would therefore be a better estimator. A key teaching point and a consequence of the Central Limit
Theorem is the approximate normal distribution of the sample mean despite the underlying lognormal distribution of the observations.

Consider the following problem:

*The authors of a paper suggest that a reasonable probability model for drill lifetime (in hours) is a lognormal distribution with $\mu = 4.5$ and $\sigma = 0.8$. a) What are the mean value and standard deviation of the drill lifetime? b) What is the probability that lifetime is at most 100? c) What is the probability that lifetime is at least 200?*

A) Use a computer to generate a single sample with $N = 100$ observations from the given distribution. Calculate estimates of the requested values. B) Use $k = 500$ repetitions to create a histogram for the sample mean. Repeat using $N = 400$.

Derive the theoretical answers and compare to your simulated results.

**Problem 2: Example Simulation Project**

\[ \text{mu} = 4.5; \text{sigma} = 0.8; N = 100; k = 500; \]

% Part A
% Generate a single sample with N observations
x = lognrnd(mu, sigma, N, 1);
% Use the sample to estimate the desired values
[mean(x) std(x) sum(x<=100)/N sum(x>=200)/N]

% Part B
% Generate k samples, each with N observations
% x is an (N x k) matrix. Each column represents a different sample.
x = lognrnd(mu, sigma, N, k);
% xm is a k element vector. Each element is the mean of a corresponding sample.
xm = mean(x);

% Create a histogram of the sample mean values
hist(xm,[2.5:5:250],1/5);
axis([0 250 0 0.08])

Program Listing 2: Simulation Code for Problem 2
Simulating Estimation Problems

Octave also makes it relatively easy to simulate problems from estimation theory. In many estimation problems the probability distribution of the estimator is unknown and simulation must be used to determine qualities of the estimator.

Problem 3 is an example estimation problem that was given as a project assignment. Students are asked to use computer simulation to compare three different estimators. The first estimator is obtained using the method of moments. The second estimator is the maximum likelihood estimator. The maximum likelihood estimator is biased. The bias can be removed by multiplying the estimator by a factor that is a function of the sample size, this yields the third estimator. The third estimator is unbiased, but it will have a greater variance than the maximum likelihood estimator (since the multiplicative factor is greater than one). Although it is possible to derive theoretical expressions for the mean value of each estimator, deriving expressions for the variance of each estimator is much more difficult. It is much simpler to use simulation to compare the estimator variances.
Octave code to create the histograms for each of the estimators is shown in Program Listing 3. The simulation code produces 40 observations for each of 500 samples and then computes estimates for each of the samples using the three different estimator formulas. Typical histograms that are produced by running the program are shown in Figures 7 - 9. The simulation can be performed using approximately 10 Octave statements. (The distribution in this problem is not a standard one. The inversion method is used to produce the observations. This requires 3 Octave statements.)

The bias of the second estimator is apparent from the histogram. (In the simulation, the true value of \( \theta \) is equal to 10, all \( \hat{\theta}_2 \) values are less than 10, so the estimator is obviously biased.) The much smaller variance of the second and third estimators compared to the first estimator is also apparent. Close inspection of the histograms shows that the variance of the third estimator is slightly greater than that of the second. If we require that estimators be unbiased, then the third estimator would be the best estimator.

**Student Survey Results**

The 19 students in the class were asked to respond to the survey shown in Table 3. Responses were on a 5 point scale with a 1 response indicating strongly agree and a 5 response indicating strongly disagree. The results are summarized using boxplots in Figure 10.

The median response to each question on the survey was 2, indicating agreement with each statement. There was little variation in the response to question 1, which implies that the students felt that the simulation projects helped them to better understand the course material. This was, of course, the hoped for response and was the main reason for adding the projects. The responses to questions 4 (previous Octave/MATLAB experience) and 5 (adequate instruction in Octave/MATLAB) had the greatest variation. This suggests that students might appreciate the projects more with additional tutorial instruction in Octave/MATLAB.

**Conclusion**

Use of simulation enhances teaching and learning of statistics. MATLAB/Octave was chosen as a simulation language for practical reasons, but the relative ease of programming is an asset. Simulations allow students to grasp concepts more firmly. They are more efficient and powerful than working with actual data sets or using repeated sampling activities. In the future I plan to provide more tutorial information on programming and increase the number of simulations during the probability portion of the course. This will help students gain familiarity with the programming environment.
Let $X_1, \ldots, X_n$ be a random sample from the following distribution:

$$f(x; \theta) = \frac{x^2}{\theta^3} \quad \text{for } 0 \leq x \leq \theta$$

where the parameter $\theta$ is unknown and is to be estimated from the sample.

The following three estimators are being proposed as estimators for $\theta$.

$$\hat{\theta}_1 = \frac{4}{3} \bar{X}, \quad \hat{\theta}_2 = \max(X_i), \quad \hat{\theta}_3 = \frac{3n+1}{3n} \max(X_i)$$

Use computer simulation to compare these three estimators. (a) Generate a sample of 40 observations from the above distribution (using “true” value of 10 for $\theta$). Use the three estimator formulas to compute estimates of $\theta$ from the sample. (b) Repeat the experiment 500 times and compute the mean and standard deviation of each of the estimators over the 500 repetitions. (c) Plot histograms of the three estimators from the 500 experiments. Which estimators appear to be unbiased? Which estimators appear to have the smallest variance? Which estimator appears to be the best estimator? Why?

Program Listing 3: Simulation Code for Problem 3

% theta is the true value of the parameter we are estimating
% n is the sample size and k is the number of repetitions of the experiment
theta = 10; n = 40; k = 500;

% Define a function that returns an (RxC) matrix of random samples from the desired
% distribution. The distribution density function is equal to:
% \[ f(x; \theta) = \frac{x^2}{\theta^3} \quad \text{for } 0 \leq x \leq \theta \]
% The inversion method is used to generate the random variables
function x = projrnd(theta, R, C)
    x = theta .* (rand(R, C).^((1/3)));
end

% Generate the observations. Observations for all k samples are stored in matrix x.
% There are n observations in each sample so x is an (n x k) matrix.
x = projrnd(theta, n, k);

% Compute the three estimates for each sample. T_est is an (k x 3) matrix.
% The three columns of T_est correspond to the three estimators.
T_est = [(4/3)*mean(x)'; max(x)'; (3*n+1)/(3*n)*max(x)'];

% Create histograms of each of the three estimators
for i = 1:3; subplot(3,1,i); hist(T_est(:,i)); end
Figure 7: Histogram of Estimator $\hat{\theta}_1$ for Problem 3

Figure 8: Histogram of Estimator $\hat{\theta}_2$ for Problem 3

Figure 9: Histogram of Estimator $\hat{\theta}_3$ for Problem 3
1. The Octave/MATLAB projects helped me to better understand the course material.
2. The number of Octave/MATLAB projects was appropriate.
3. The level of difficulty of the Octave/MATLAB projects was appropriate.
4. I had used Octave/MATLAB prior to using it in this course.
5. I felt that I had adequate instruction in Octave/MATLAB in order to complete the projects.
6. I would recommend that Octave/MATLAB projects be used in this course in the future.

Table 3: Student Survey

![Survey Results](image)

Figure 10: Survey Results (1 – Strongly Agree, 5 – Strongly Disagree)

**Bibliography**